Labeling of data-driven complexes for surface reconstruction

Patrick Labatut

École Doctorale de Sciences Mathématiques de Paris Centre

Laboratoire IMAGINE (École des Ponts/CSTB)

Doctorat

Informatique

Partition de complexes guidés par les données pour la reconstruction de surface

Patrick Labatut

Thèse dirigée par Renaud Keriven

Soutenue le 14 septembre 2009

Pierre Alliez	Rapporteur
Christian Choffrut	Examinateur
Olivier Faugeras	Examinateur
Renaud Keriven	Directeur
Marc Pollefeys	Rapporteur
Jean-Philippe Pons	Co-directeur

École Doctorale de Sciences Mathématiques de Paris Centre

Laboratoire IMAGINE (École des Ponts/CSTB)

Doctorate of Science

Computer Science

Labeling of data-driven complexes for surface reconstruction

Patrick Labatut

Under the supervision of Renaud Keriven

Thesis defended on September 14, 2009

Pierre Alliez	Reviewer
Christian Choffrut	Examiner
Olivier Faugeras	Examiner
Renaud Keriven	Advisor
Marc Pollefeys	Reviewer
Jean-Philippe Pons	Co-advisor

Résumé

Titre Partition de complexes guidés par les données pour la reconstruction de surface

Résumé Cette thèse introduit une nouvelle approche pour la reconstruction de surface à partir d'acquisitions de nuages de points. Cette approche construit un complexe cellulaire à partir du nuage de points puis formule la reconstruction comme un problème d'étiquetage binaire des cellules de ce complexe sous un ensemble de contraintes de visibilité. La résolution du problème se ramène alors au calcul d'une coupe minimale *s-t* permettant d'obtenir efficacement une surface optimale d'après ces contraintes.

Dans la première partie de cette thèse, l'approche est utilisée pour la reconstruction générique de surface. Une première application aboutit à un algorithme très robuste de reconstruction de surface à partir de nuages denses issus d'acquisitions laser. Une seconde application utilise une variante de cet algorithme au sein d'une chaîne de photo-modélisation en combinaison avec un raffinement variationnel photométrique. La chaîne complète est adaptée à la reconstruction de scènes de grande échelle et obtient d'excellents résultats en terme de complétude et de précision des reconstructions.

La seconde partie de cette thèse considère le problème de la reconstruction directe de modèles géométriques simples à partir de nuages de points. Un algorithme robuste est proposé pour décomposer hiérarchiquement des nuages de points denses en formes issues d'un ensemble restreint de classes de formes. Lorsque que cet ensemble de classes est réduit aux plans seulement, la reconstruction de modèles de très faible complexité est possible. Une extension à d'autres classes de formes échange cet avantage contre la gestion de nuages de points plus difficiles.

Mots clés nuage de points, reconstruction de surface, complexe cellulaire, optimisation discrète, coupe minimale s-t, visibilité, triangulation de Delaunay, complexe polyhédral, BSP, robustesse, données laser, stéréovision multivues, photo-modélisation

Abstract

Title Labeling of data-driven cell complexes for surface reconstruction

Abstract This thesis introduces a new flexible framework for surface reconstruction from acquired point sets. This framework casts the surface reconstruction problem as a cells binary labeling problem on a point-guided cell complex under a combination of visibility constraints. This problem can be solved by computing a simple minimum s-t cut allowing an optimal visibility-consistent surface to be efficiently found.

In the first part of this thesis, the framework is used for general surface reconstruction problems. A first application leads to an extremely robust surface reconstruction algorithm for dense point clouds from range data. A second application consists in a key component of a dense multi-view stereo reconstruction pipeline, combined with a carefully designed photometric variational refinement. The whole pipeline is suitable to large-scale scenes and achieves state-of-the-art results both in completeness and accuracy of the obtained reconstructions.

In the second part of this thesis, the problem of directly reconstructing geometrically simple models from point clouds is addressed. A robust algorithm is proposed to hierarchically cluster a dense point clouds into shapes from a predefined set of classes. If this set of classes is reduced to planes only, the concise reconstruction of models of extremely low combinatorial complexity is achieved. The extension to more general shapes trades this conciseness for a more verbose reconstruction with the added feature of handling more challenging point clouds.

Keywords point cloud, surface reconstruction, cell complex, discrete optimization, minimum s-t cut, visibility, Delaunay triangulation, polyhedral complex, BSP, robust, range data, multi-view stereo, image-based modeling

Remerciements

Je tiens d'abord à exprimer toute ma reconnaissance à mon directeur de thèse, Renaud Keriven, pour m'avoir accueilli au sein de son laboratoire et avoir dirigé les travaux présentés dans ce manuscrit. C'est très certainement en assistant à ses cours que son enthousiasme communicatif pour le domaine de la vision par ordinateur m'a convaincu de poursuivre dans cette voie. J'adresse également toute ma gratitude à Jean-Philippe Pons qui a fortement influencé la direction prise par cette thèse.

Je mesure l'honneur que me font les membres du jury par leur présence le jour de la soutenance. En particulier, je remercie Pierre Alliez et Marc Pollefeys pour avoir accepté d'endosser le rôle de rapporteur malgré le travail important que cela constitue. Je remercie aussi Olivier Faugeras et Christian Choffrut pour leur participation au jury en tant qu'examinateur.

Mes remerciements vont bien sûr aux différents membres de l'équipe IMAGINE, de l'ancien laboratoire CERTIS et de l'ancienne équipe Odyssée que j'ai pu rencontrer, avec qui j'ai pu collaborer ou simplement discuter.

Je dois finalement remercier mes amis et ma famille qui ont su à la fois m'apporter le soutien et les distractions nécessaires pour mener à terme cette thèse.

Contents

Notations

$\mathbb{E}^d$	the Euclidean space of dimension d
$\mathcal{C}$	a cell complex
$\mathcal{C}_k$	the set of k-faces of a cell complex
C	a cell in a cell complex, *i.e.* a d-face in dimension d
F	a facet in a cell complex, *i.e.* a $(d-1)$-face in dimension d
$F^{C_i \to C_j}$	a facet F oriented from cell C_i to cell C_j
$\mathcal{S}$	a pseudo-surface
l	a binary labeling of cells
l_C	the label of the cell C
$[\![i, j]\!]$	the set of integers $\{i, i+1, \ldots, j-1, j\}$
$\mathbb{1}[.]$	the indicator function (1 if the predicate . is true, 0 else)
$\mathcal{P}$	a set of "acquired" points in $\mathbb{E}^3$
P	a point in $\mathbb{E}^3$
v_P	the set of ending points of the line-of-sights associated with an "acquired" point P in $\mathbb{E}^3$
$\mathbb{R}^d$	the real space of dimension d
$\boldsymbol{x}$	a point in $\mathbb{R}^d$
x	a scalar in $\mathbb{R}$
$*$	the convolution product, *i.e.* : $(f * g)(x, y) = \int_{\mathbb{R}^2} f(u, v)\, g(x-u, y-v)\, \mathrm{d}u\mathrm{d}v$ for two functions f and g defined on $\mathbb{R}^2$
$\otimes$	the tensor product: $\mathbb{R}^d \times \mathbb{R}^d \to \mathcal{M}_{d,d}(\mathbb{R})$
$\mathrm{Diag}(M)$	the matrix whose diagonal coefficients are the same as M and extra-diagonal coefficients are zero
$\overrightarrow{u} \times \overrightarrow{v}$	the cross-product of two vectors $\overrightarrow{u}$ and $\overrightarrow{v}$ in $\mathbb{E}^3$
$\overrightarrow{u} \,.\, \overrightarrow{v}$	the scalar product of two vectors $\overrightarrow{u}$ and $\overrightarrow{v}$ in $\mathbb{E}^3$
$\mathrm{aff}(U)$	the affine hull of the set U
$\mathrm{conv}(U)$	the convex hull of the set U

Introduction

Computer vision tries to make computer see and understand visual data like humans. On the contrary, computer graphics deals with the synthesis and manipulation of visual content from already digitized data. While these objectives might seem completely opposite, the two computer science research fields are growingly borrowing from each other. The particular problem of modeling objects or scenes from measurements has been, in both fields, one of the most studied problem. In computer graphics, these measurements come directly from points acquired with range scanning devices and in computer vision, they come from images and are extracted with less accurate matching techniques.

Motivation

The automatic reconstruction of computer representations of scenes and objects has numerous applications in various domains and has thus always motivated research in both computer graphics and vision. In computer aided geometric design (CAGD), the reconstruction from range-acquired point clouds sampling an existing object is referred as reverse-engineering and is used to replace physical prototypes. Medical applications include computer aided diagnostic, therapy and surgery planning and monitoring and require modeling organs and tissues of the human body from 2D or 3D images or scans. The reconstruction problem is also relevant for natural sciences like geology and topography where not only the visualizations of large-scale environments is needed but also actual numerical models to be used for simulations. In the history and art field, the digital archival of cultural heritage not only allows the creation of virtual museums for culture diffusion but facilitates restoration, preservation and understanding of the artist work. Finally the entertainment industry through video games and movies has become one

of the most active demander for reconstructions to reproduce reality from scratch for improved realism.

High-precision multi-view stereo

The modeling of small objects from a set of calibrated (and silhouetted) images taken under controlled imaging conditions triggered several radically different approaches in early multi-view stereo methods: from greedy optimization [Kutulakos and Seitz, 2000] to deformable models [Faugeras and Keriven, 1998] and graph cuts [Snow et al., 2000]. The field has quickly matured in the recent years thanks to the stimulating competition proposed in [Seitz et al., 2006]. The results of the latest methods are getting closer to the range measurements used as a reference. However, few of the proposed methods for multi-view stereo are readily applicable to large-scale scenes.

Large-scale multi-view stereo

To overcome scalability issues and the lack of silhouettes in outdoor scenes, the few still applicable approaches rely on sparse data measurements, either as depth maps [Strecha et al., 2006, Goesele et al., 2007] or filtered and expanded surface patches [Furukawa and Ponce, 2007] that are merged to produce a surface after applying a standard surface reconstruction algorithm such as [Kazhdan et al., 2006]. The performance for complete reconstruction seems to be lower than for smaller-scale objects, either in accuracy or completeness of the obtained model. Handling visibility globally and consistently is indeed a problem in the merging step.

Dealing with visibility

Early methods in multi-view stereo [Kutulakos and Seitz, 2000] and reconstruction from range scans [Curless and Levoy, 1996] acknowledged the importance of consistent visibility to guide the reconstruction. In both fields, the focus has shifted towards accurate reconstructions. In multi-view stereo where measurements are less dense and cues guiding the reconstruction seem more important, the later methods developed for small-scale objects either assumed the visibility of the surface to be given [Vogiatzis et al., 2005] or in deformable models, the information was iteratively estimated [Pons et al., 2007a].

Robustness issues in surface reconstruction

Proper modeling of the acquisition process to guide the reconstruction can strongly contributes to the robustness. The issue of robustness to high level of noise and outliers is well known in computer vision where slight errors

mismatches in images are hard to avoid. Most surface reconstruction algorithms from point clouds in computer graphics implicitly assume input point clouds of range-scanning quality: dense, almost uniformly sampled and outlier-free. This lack of versatility limits the use of such methods in surface reconstruction for computer vision applications.

Reconstruction of compact models

While surface reconstruction from point clouds aims at producing precise reconstructions, modeling very fine details necessitates overly complex models which are often not practical not only for visualization but also for higher-level applications such as indexing and recognition. More lightweight and geometrically simpler representations are highly desirable. Such simplified models are currently built using either very problem-specific methods or with extensive user intervention (related techniques are surveyed in Chapter 5).

Thesis outline

Chapter 1 first deals with the practical details of the acquisition of point clouds, either in the case of triangulation or time-of-flight range scanners, or in the case of passive stereo. The two point cloud generation algorithms used for multi-view passive stereo in the subsequent chapters are also fully described.

Chapter 2 presents a thorough review of existing surface reconstruction algorithms from point clouds. Following Chapter 1, a visibility-based surface fitting framework on point-driven cell complexes is introduced. In particular, the original visibility term upon which all our subsequent surface reconstruction algorithms are based is proposed and discussed along with the corresponding optimization method. This surface fitting framework is also compared to other closely related methods.

The first part of this thesis handles two major challenging surface reconstruction problems in computer graphics and computer vision.

Chapter 3 considers the problem of surface reconstruction from range data. The framework of Chapter 2 is used on the simplicial complex of the Delaunay triangulation. The visibility term is relaxed to accommodate for both the density of range point clouds and the scarcity of line-of-sight information. Derivated from a well-known geometric criterion for curve reconstruction, an additional surface quality term is also proposed. The robustness of the whole approach is demonstrated through numerous experiments with several other methods: the method is shown to compare favourably with the state of the art when dealing with significantly altered data sets as inputs.

The visibility-based surface optimization framework introduced in Chapter 2 contains key components to address some of the major flaws of existing dense multi-view stereo methods. An analysis of the weakness of existing

dense multi-view stereo algorithms for large-scale scene reconstructions is done. By relying again on a Delaunay triangulation, we present another point-based surface reconstruction algorithm for multi-view stereo which permits the reconstruction of an initial visibility-consistent mesh from multi-view stereo point clouds with typically important levels of noise and high outlier ratios. A carefully designed variational photo-refinement is used as a post-processing to significantly improve the precision of this initial coarse surface. The whole multi-view stereo pipeline combining the point cloud generation, the computation of an initial complete mesh and the final mesh refinement overcomes all the mentioned shortcomings while still being highly accurate. It is evaluated on several challenging outdoor large-scale scenes and compared with other dense multi-view stereo methods also suitable for large-scale scenes. It is demonstrated to consistently compare very favorably with the competition on both completeness and accuracy of the reconstruction.

The second part of this thesis tackles the problem of concise or geometrically simplified modeling from point clouds.

In Chapter 5, existing techniques that possibly relates to geometrically simplified modeling are surveyed. The conclusion is drawn that neither in the context of multi-view stereo nor for reconstruction from range data an appropriate method for surface reconstruction from point clouds with strong shape priors exists or could satisfyingly be obtained by combining existing techniques. An overview of our original general approach to simplified modeling from point clouds is presented and its various benefits are discussed: briefly, after detecting various shapes in a point cloud, these shapes are embedded in the optimization domain, and a segmented piecewise-primitive surface is reconstructed.

Chapter 6 describes the first step of our simplified modeling technique from point clouds which extracts shapes from point clouds. After an introduction to various robust regression methods suitable for multiple shape detection, a hierarchical clustering algorithm is proposed. Single shape extraction is done by adapting the random sampling framework to deal with quasi-dense point clouds, and multiple shapes are extracted hierarchically to decompose a sufficiently dense point cloud into a set of simple shapes from predefined classes of shapes.

Concise reconstructions are introduced in Chapter 7. From a hierarchical decomposition of point clouds in purely planar regions, the proposed approach outputs a reconstructed surface capturing the most prominent planar features of an object or a scene in a not only geometrically simple model but also a lightweight reconstruction with an extremely low combinatorial complexity.

The approach of Chapter 7 is generalized to allow the reconstruction of piecewise-primitive models assembling patches from a set of selected classes of second order surfaces. While the conciseness of the reconstructions of Chapter 8 is traded for the accurate modeling of more complex shapes, the

surface reconstruction is also extended to handle difficult point clouds in which severe occlusions or insufficient sampling limit the efficiency of the shape extraction and the subsequent robustness of the surface reconstruction step.

Finally, Chapter 8.2 summarizes the various contributions of this thesis, the benefits of the proposed approaches, discusses their limitations and proposes some possible tracks for future work.

To conclude, let us note that part of the material herein comes from published papers, papers pending publication or in preparation:

- the visibility term presented in Chapter 2 was initially introduced over Delaunay triangulations in *Efficient Multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and graph cuts* [Labatut et al., 2007] with Jean-Philippe Pons and Renaud Keriven in the proceedings of the IEEE International Conference on Computer Vision, 2007;
- the range surface reconstruction algorithm of Chapter 3 will appear in the journal Computer Graphics Forum under the title *Robust and efficient surface reconstruction from range data* [Labatut et al., 2009a] with Jean-Philippe Pons and Renaud Keriven;
- the whole large-scale multi-view stereo pipeline of Chapter 4 is joint work with Jean-Philippe Pons, Renaud Keriven and Vu Hoang Hiep and appears in *Towards high-resolution large-scale multi-view stereo* [Vu et al., 2009] with Vu Hoang Hiep, Renaud Keriven and Jean-Philippe Pons in the proceedings the IEEE Conference on Computer Vision and Pattern Recognition, 2009. A journal paper entitled *Global and visibility-consistent dense multi-view stereo for large-scale scenes* with Jean-Philippe Pons, Renaud Keriven and Vu Hoang Hiep, describing the pipeline in further details is in preparation;
- the hierarchical shape extraction and shape-based surface reconstruction of Chapter 6 and Chapter 8 will appear in *Hierarchical shape-based surface reconstruction for dense multi-view stereo* [Labatut et al., 2009b] with Jean-Philippe Pons and Renaud Keriven in the proceedings of the 2009 IEEE International Workshop on 3-D Digital Imaging and Modeling.

CHAPTER 1

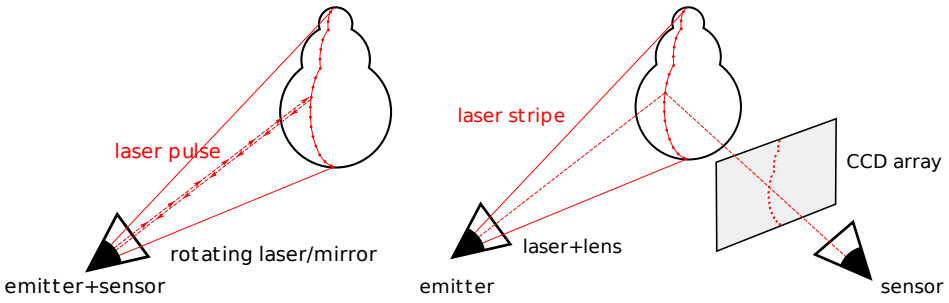
Acquisition of point-sampled geometry

This chapter describes the different acquisition processes considered in this thesis to get point-sampled geometry. First, a quick overview of range scanning techniques is given and then, two different approaches to compute quasi-dense point cloud from calibrated images are presented. The common availability of lines of sight information associated to these point clouds is underlined. This information is instrumental in the visibility-based surface fitting framework presented in the next chapter.

Contents

As reminded in the introduction, surface acquisition has numerous applications for instance in engineering (reverse-engineering, prototyping or visualisation), in the entertainment industry for video games and movies, and in the digital archival for cultural heritage preservation. Several devices and acquisition processes suitable to different scenarios have thus been conceived. In this chapter, we present and discuss the various techniques used to acquire point-sampled geometry in this thesis. Since this thesis focuses on surface reconstruction algorithms, only geometry acquisition and not appearance acquisition will be described.

1.1 Point sets from laser range data



(a) A time-of-flight range finder (b) A triangulation range finder

Figure 1.1: ***The two common types of laser range finders.***

Range finding is perhaps the most used form of acquisition of point-sampled geometry in practice. It consists in measuring the distance between the device and the target object. Two typical setups exists and are depicted in Figure 1.1.

The first setup shown in Figure 1.1(a), for time-of-flight range finders, pairs together the emitter and the sensor and measures the distance between an emitter and the target object by firing a short pulse of light (on the order of picoseconds) and timing how long it takes to return. SONAR and RADAR devices exploit a similar concept, but rely respectively on sound propagation and microwaves or radio waves and are better suited to very large-scale acquisitions. Of more interest here, LIDAR (LIght Detection And Ranging) is designed for smaller scale acquisitions. The acquisition is often done by sweeping a whole angular sector and generating a stripe of depth measurements. However it typically requires long scanning times and its absolute accuracy is generally poor.

The second common type of range finger setup, shown in Figure 1.1(b), separates the emitter and sensor. It is based on active optical triangulation:

the emitter illuminates the target object with a narrow beam or a whole light stripe and the corresponding illuminated positions on the object are recorded by the sensor (which is either a photodiode or a CCD array). Since the distance between the emitter and sensor is known along with the relative orientation of the sensor and the direction of the emitter, a depth can be measured and the relative location in 3D of the point on the object can be inferred. Several acquisitions are commonly done in a row by translating the emitter-sensor pair relatively to the object to acquire a batch of range stripes that are assembled as a single range image. Other setups for batch acquisitions are also possible (rotating the object or the acquisition system or translating only the emitter or the sensor). The acquired range image can be thought as being imaged at once by a pushbroom depth camera[1] that is often approximated by a simple orthographic camera[2]. The absolute accuracy of such scanners is typically much higher than that of time-of-flight scanners but they are mainly restricted to small-scale objects.

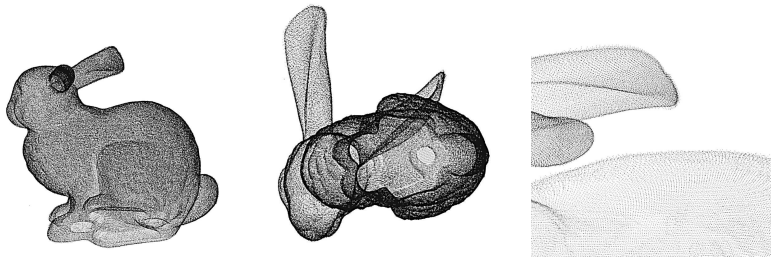
To give some actual numbers, the triangulation range scanning device used to acquire the ground truth models in the dense multi-view stereo benchmark of [Seitz et al., 2006] (a Cyberware Model 15) has a depth accuracy of about 0.05mm to 0.2mm with a resolution of 0.25mm and the scanner used in the Stanford 3D scanning repository [sta, 2009] (a Cyberware 3030 MS) has similar specifications (see Figure 1.2 for sample scans). In comparison, the expected accuracy of the LIDAR device used in the dense multi-view stereo benchmark of [Strecha et al., 2008] can be of several millimeters. The size of the scanned object or scene is however significantly different: at most a couple dozens centimeters for the triangulation scanners, and from one to several dozens meters for the LIDAR laser, so that while the absolute accuracy is lower the relative accuracy is still comparable (see Figure 1.3 for an example of mobile LIDAR scans).

Range stripes or range images may be acquired independently, but since the acquisition (and subsequent surface reconstruction) of a whole object or scene is expected, a registration step that aligns in the same coordinate system the different range stripes or images with rigid-body transformations is needed. For time-of-flight range finder, this step is often done implicitly by the device during the scanning. As an example, Chapter 3 shows results on a data set acquired using a car driving in a street, and the whole acquisition unit features a time-of-flight range finder paired with a GPS/IMU making possible the localisation and registration of the successive range stripes in real-time. For triangulation range finders, the registration of the different images is done either also automatically or with a variant of the classical Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992].

[1] A camera whose restriction to the plane formed by the range stripe and the emitter is a perspective camera and whose optical center follows the translation of the emitter.

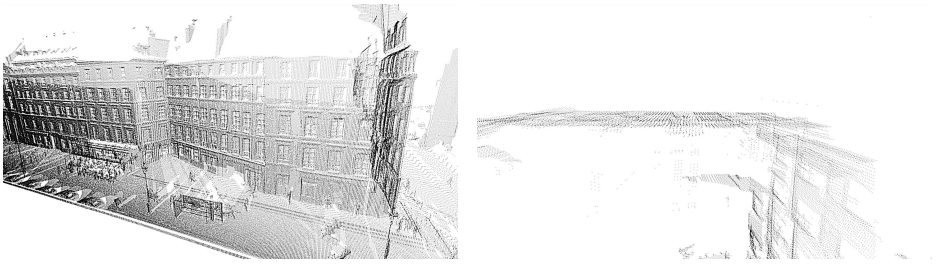
[2] A perspective camera whose optical center is rejected at infinity along the optical axis.

(a) A single range image (b) A single range image (top view) (c) A single range image (close-up view)



(d) Merged range images (e) Merged range images (top view) (f) Merged range images (close-up view)

Figure 1.2: ***Stanford bunny**: a point cloud from a triangulation range scanner.*



(a) Merged range stripes (b) Merged range stripes (close-up view)

Figure 1.3: ***rue Soufflot**: a point cloud from a mobile time-of-flight range finder.*

According to [Curless, 1997] and for triangulation range fingers, range errors dominate alignments errors. The range error noise distribution can be considered as anisotropic along the emitter line-of-sight and the range uncertainties may be modeled with simple Gaussian statistics. A difficulty for surface reconstruction algorithms lies in the fact that the noise level is typically in the order of the sample spacing.

In addition to simple noise, errors in range measurements may have other origins: errors reflectance discontinuities, shapes variations (corners), sensor

occlusions (portion of the light block from the sensor) and laser speckle interference due to surface roughness. A thorough analysis of these different sources of measurements errors in triangulation range scanners can be found in [Curless, 1997].

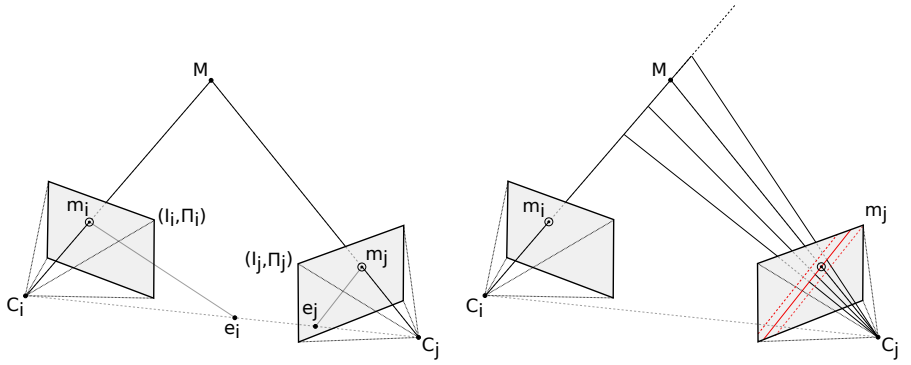
To conclude this section, while range scanning devices allow measurements of much greater quality in terms of density and uniformity of sampling than what is achievable with the simple passive image-based approaches described in the next section, they also have a number of limitations:

- the scanning device is often expensive with a price several orders of magnitude higher than a standard consumer camera used in passive stereo,
- the acquisition process can be lengthy and necessitate tedious setups especially for the acquisition of large-scale scene. In [Banno et al., 2008], the use of flying balloons with attached range sensors to acquire scans of the Bayon temple is described along with the subsequent error correction and registration methods,
- the number of generated points is huge: the acquisition of 10^6 to 10^9 samples is becoming commonplace. This often demands dedicated algorithms to either first downsample the point sets to make the surface reconstruction more tractable or sophisticated out-of-core or streaming reconstruction algorithms to handle such amount of data with a reasonable memory footprint.

1.2 Point sets from multi-view passive stereo

An object or a scene can also be point-sampled using passive stereo techniques. Taking a few pictures of a scenes is not sufficient to relate corresponding points in different images. A first problem consists in determining the "Structure from Motion" (abbreviated SfM): by using only the acquired images, find the calibration of the cameras *i.e.* their relative position and their internal settings (focal length, etc...). In this thesis, we assume that the cameras have already been fully calibrated. The underlying optimization problems are well studied and have been demonstrated in [Pollefeys et al., 2004, Brown and Lowe, 2005, Martinec, 2008]. Details on the principles can be found in computer vision reference books [Faugeras et al., 2001, Hartley and Zisserman, 2004]. More recently and on a larger scale, a working combination of these techniques was shown in [Snavely et al., 2008]. SfM approaches robustly extract and track sparse matched points in images, compute initial camera poses and 3D points position and optimize both for camera parameters and 3D positions of the tracked features with a "bundle adjustment" [Triggs et al., 2000], a large non-linear least-squares optimization.

In the sequel, n views are given, each with a corresponding image I_i :



(a) The epipolar geometry of two cameras Π_i and Π_j (with cameras optical centers C_i and C_j and epipoles e_i and e_j) and the projections m_i and m_j of a point M.

(b) A candidate match for a point can be sought either in a small band around the epipolar lines (for interest points) or by plane sweeping.

Figure 1.4: ***A stereo camera pair.***

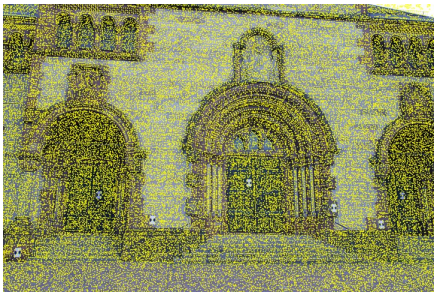
$\mathbb{R}^2 \rightarrow \mathbb{R}$ ($\mathbb{R}^3$ for color images) and a camera Π_i which maps a 3D point $M \in \mathbb{R}^3$ to a 2D point m the coordinate frame of the image I_i. Since the input images are also assumed to have been corrected for radial distortion, and that the cameras are modeled with a simple perspective or pinhole camera model, Π_i is a perspective projection described by a 3×4 homogeneous matrix Π_i and $\tilde{m} = \Pi_i \begin{pmatrix} M & 1 \end{pmatrix}^T = (\tilde{m}_x\, \tilde{m}_y\, \tilde{m}_z)^T$ in homogeneous coordinates or $m = \begin{pmatrix} \tilde{m}_x/\tilde{m}_z\, \tilde{m}_y/\tilde{m}_z \end{pmatrix}^T$ in Cartesian coordinates. Given a point $\tilde{m}_i$ in one image I_i, the corresponding point in the image I_j is located along the *epipolar line* of $\tilde{m}_i$ (see Figure 1.4(a)), which is the projection in the coordinate frame of the image I_j of the ray from the optical center C_i of the camera i to the point $\tilde{m}_i$ and is given by the relation $\tilde{m}_i^T\, F_{ij}\, \tilde{m}_j = 0$ where F_{ij} is the fundamental matrix F_{ij} of the oriented camera pair (i, j) Faugeras et al. [2001]. Besides camera calibration, the particular choice of camera pairs is also given: as images are typically acquired in sequence, it can generally be easily inferred.

Two different but related point cloud generation techniques are now described, one matching interest points in the input images and another using plane sweeping to compute sparse depth maps. In order to apply the surface fitting framework of Chapter 2, a slightly non-conventional way to generate point clouds from passive stereo is used that favors density over matching robustness. Finally, the preliminary point cloud generation of our work in [Labatut et al., 2007] relied on SIFT keypoints and descriptors [Lowe, 2004], an approach that presented several drawbacks: first, due to the invariance over scale and rotation, high density is more difficult to achieve, also the

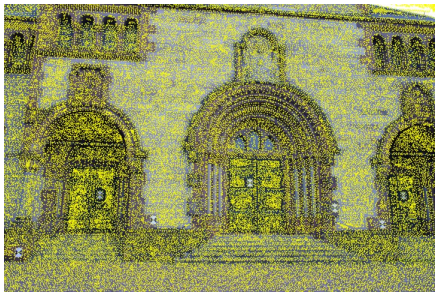
epipolar geometry which is known but not used at all in the descriptor computation.

1.2.1 Interest points

(a) An image from the *Herz-Jesu-P25* data set of [Strecha et al., 2008]



(b) extracted Harris corners



(c) extracted LoG blobs

Figure 1.5: ***Interest points.*** *Both Harris corners and LoG blobs are extracted from the input images at a small scale to try capturing most of the geometry.*

First, interest points are located in all the input images. To this purpose, and to capture most of the geometry of the sampled shape (see Figure 1.5), two complementary kinds of interest points are considered: Harris corners which are typically lying on "corners" in images and Laplacian-of-Gaussian located at the center of blob-like structures in images. The definitions and origins of these interest points are first briefly recalled.

Harris corners

Let $I : \mathbb{R}^2 \to \mathbb{R}$ be a grayscale image. We note G^σ the standard two-dimensional Gaussian kernel of variance σ^2:

$$G^\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} ,$$

and L^σ the scale-space representation of the image I obtained by convolution with G^σ: $L^\sigma = G^\sigma * I$ to detect interest point at the scale σ [Lindeberg, 1998]. We consider a translated version K^σ of the image L^σ by a vector (u, v): $\forall (x, y) \in \Omega$, $K^\sigma(x, y) = L^\sigma(x+u, y+v)$. To integrate over a weighted neighborhood of a point (x_0, y_0), a Gaussian window with an integrative scale τ is introduced. The two scales σ and τ are usually linked by a relation of the form $\sigma = \xi\tau$ where ξ typically is in $[0.5, 0.7]$. Estimating the correspondingly weighted sum of squared differences $S^{\sigma,\tau}(u, v)$ between L^σ and K^σ can be done by approximating J with a first order Taylor expansion of $I(x+u, y+v)$ and gives $S^{\sigma,\tau}(u, v) \approx \begin{pmatrix} u & v \end{pmatrix} M^{\sigma,\tau} \begin{pmatrix} u \\ v \end{pmatrix}$ with $M^{\sigma,\tau}$ the scaled *structure tensor* or second-moment matrix:

$$M^{\sigma,\tau} = G^\tau * \begin{pmatrix} L_x^{\sigma 2} & L_x^\sigma L_y^\sigma \\ L_x^\sigma L_y^\sigma & L_y^{\sigma 2} \end{pmatrix} ,$$

where $L_x^\sigma = \frac{\partial L^\sigma}{\partial x}$, $L_y^\sigma = \frac{\partial L^\sigma}{\partial y}$. Large variations of $S^{\sigma,\tau}(u, v)$ in all directions (u, v) characterize corners at an image scale σ. If the matrix $M^{\sigma,\tau}$ has two small eigenvalues λ_1 and λ_2, then the considered point (x_0, y_0) is not an interest point. If the matrix $M^{\sigma,\tau}$ has a small and a large eigenvalue, the point is located on an edge at a scale σ. Finally two large eigenvalues of $M^{\sigma,\tau}$ correspond to a corner point at (x_0, y_0). Instead of computing explicitly the eigenvalues of $M^{\sigma,\tau}$ to find corners, strong local maxima[3] of the more lightweight Harris corner measure $h(M^{\sigma,\tau})$ [Harris and Stephens, 1988] are used:

$$h(M^{\sigma,\tau}) = \det(M^{\sigma,\tau}) - \kappa \operatorname{Tr}(M^{\sigma,\tau})^2 = \lambda_1\lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 ,$$

where κ is a sensitivity parameter usually ranging from 0.04 to 0.15.

Laplacian-of-Gaussian blobs

The zero crossing of the Laplacian Δ are often used to detect edges in an image. To detect blobs instead, the Laplacian-of-Gaussian (abbreviated as LoG) applies the Laplacian Δ operator to a scale-space representation L^σ of an image I:

$$\Delta L^\sigma = L_{xx}^\sigma + L_{yy}^\sigma = \Delta (G^\sigma * I) = (\Delta G^\sigma) * I .$$

And strong local extrema of the LoG ΔL^σ can be used to locate potential blobs: strong positive responses correspond to dark blobs and strong negative responses to bright blobs at a scale σ.

[3] local maxima above some fixed threshold value.

Matching interest points

As announced, LoG blobs and Harris corners are extracted at some fixed scale[4] in all the input images. Then, for each potential camera pair (i, j) and for each interest point m_i (of the same type) in the first image I_i of this pair, its best matching point $m_j^\star$ is sought within a small band around the corresponding epipolar line in the other image I_j (see Figure 1.4(b)). The width of this band is fixed and should partially depend on the accuracy of the calibration[5].

The best matching point $m_j^\star$ is the point with the highest matching score against the reference interest point m_i. The neighborhood of a potential match m_j in the image I_j is reprojected in the reference image I_i through a plane parallel to the focal plane of the camera Π_i and passing through the potential reconstructed 3D point (the underlying assumption is that the surface is locally fronto-parallel to the camera Π_i). The matching score can then be estimated in a window around the reference point. Since the choice of an appropriate matching window size is difficult, multi-level matching is used, and the matching criterion is the sum of normalized cross correlations (NCC) for several fixed window sizes[6] (or scale σ) as in [Yang and Pollefeys, 2003]. Also the matching windows are chosen as smooth Gaussian windows to lessen the effects of the fronto-parallel assumption. Finally the matching score $\rho(m_j)$ is:

$$\rho(m_j) = \sum_\sigma \mathrm{NCC}^\sigma(I_i, I_j \circ \Pi_j \circ \Pi_i^{-1})(m_i) \ ,$$

with:

$$\begin{aligned} \mathrm{NCC}^\sigma(I,J) &= \nu^\sigma(I,J) \,/\, \sqrt{\nu^\sigma(I)\,\nu^\sigma(J)} \ , \\ \nu^\sigma(I,J) &= G^\sigma * (I - \mu^\sigma(I))(J - \mu^\sigma(J)) \,/\, \omega^\sigma \ , \\ \nu^\sigma(I) &= G^\sigma * (I - \mu^\sigma(I))^2 \,/\, \omega^\sigma + \tau^2 \ , \\ \mu^\sigma(I) &= G^\sigma * I \,/\, \omega^\sigma \ , \end{aligned}$$

where NCC^σ is the whole image normalized cross-correlation over a Gaussian window of scale σ, $\Pi_j \circ \Pi_i^{-1}$ denotes the fronto-parallel reprojection from image I_i to I_j, $\omega_\sigma(x) = \int_\Omega G_\sigma(x - y)\, dy$ is used to normalize the involved quantities with respect to the shape of the correlation window and τ is a small constant to avoid numerical problems.

Furthermore, this best matching interest point $m_j^\star$ is kept only if its matching score $\rho(m_j^\star)$ is above some threshold and if it is also successfully

[4] in practice, a scale of 2 pixels is used for 6 Mpix images.
[5] a 3 pixel-wide band is typically chosen for 6 Mpix images.
[6] 5 levels are used on 6 Mpix images.

validated: the original interest point has to be the best matching interest point of its best matching interest point. An initial 3D point can then be reconstructed from the calibration and using standard triangulation optimization [Hartley and Zisserman, 2004] by minimizing its squared Euclidean distance to the rays (a non-linear least-squares optimization) or with a simpler iteratively reweighted optimization using the point inverse depths.

The final step aggregates the different 3D points. In each image, the 2D Delaunay triangulation (Appendix B gives a definition) of the interest points (of the same type) is computed. This geometric data structure allows to efficiently locate the nearest interest points of a given 2D point. Now, a pair of matched interest points in two different views has given rise to a 3D point by triangulation. By projecting this initial 3D point in the other views, potential other unmatched interest points that are close enough (within a tolerance similar the half-width of the epipolar band) are located. Closest unmatched interest points are merged with the original pair and a new 3D point (replacing the previous one) is re-estimated from all the interest points. The final result is a set of points each carrying a tuple of views where they were seen. Additionally, a confidence value has been assigned to each 3D point, cumulating the photo-consistency scores of all its originating pairs. Obviously, as the whole technique relies on simple greedy or winner-take-all "optimizations", it possibly generates a noisy point cloud with a decent amount of outliers.

1.2.2 Sparse depth maps

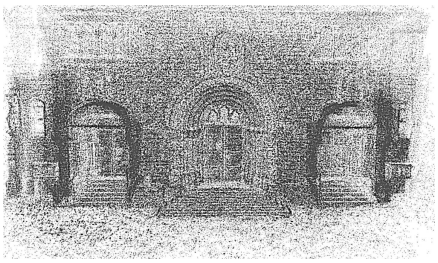
While the previous passive stereo approach is general and copes with scenes that have enough texture, it tends to generate lots of outliers and the 3D points are often poorly located. A different passive stereo technique can be devised when strong planar structures are observed as is often the case in architectural scenes.

Initial sparse depth maps are computed between pairs of input images. These depth maps have a downscaled resolution[7] *w.r.t.* the images and are filled using a simple geometric plane sweep with the same thresholded multi-level NCC matching score as above. A plane is swept in the reference camera frustum and its offset follows a geometric sequence between the near and far planes of the camera (see Figure 1.4(b)).

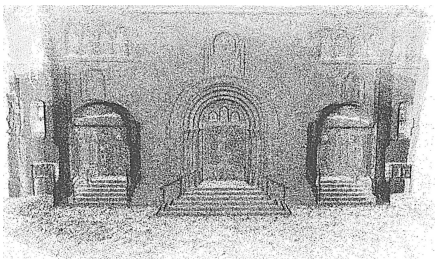
These initial depth maps are merged and clusters of points are formed according to their position in the different camera frustums. These clusters are hierarchically split until the bounding boxes of their projections in the images is small enough. A 3D k-D tree [Bentley, 1975] of this clustered initial point set is then build to efficiently find the k nearest neighbors of each point using a large neighborhood[8]. A plane is tentatively fitted to each

[7] by a factor 4×4 for 6 Mpix images.
[8] $k = 25$.

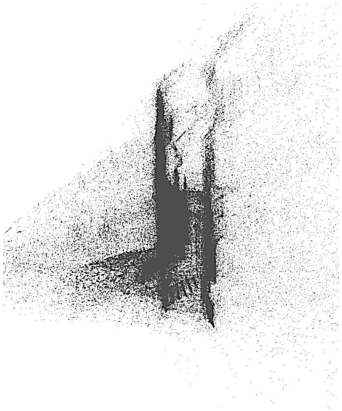
(a) One of the images of the *Herz-Jesu-P25* data set



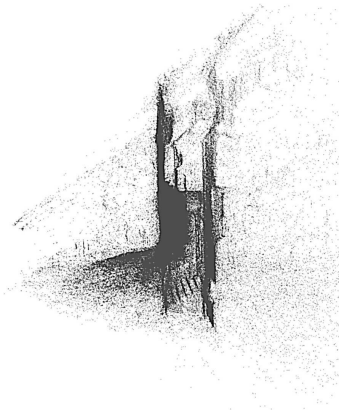
(b) Point cloud from interest points



(c) Point cloud from sparse depth maps



(d) Point cloud from interest points (side view)



(e) Point cloud from sparse depth maps (side view)

Figure 1.6: ***A visual comparison of point clouds*** *from the two passive multi-view stereo techniques on the* Herz-Jesu-P25 *of [Strecha et al., 2008].*

point's neighborhood with least-squares. Provided the fit is good enough, the point is retained and its position is iteratively refined using the same matching score as above. The final result is the same as what was obtained from interest points: a set of points each carrying a tuple of views where they were seen and an associated confidence. Again, this step still generates a noisy point cloud with a decent amount of outliers but tends to yield better

results on architectural scenes (less outliers and noise).

An advantage of the two presented passive stereo techniques lies in the fact that the reprojection and multi-level matching process can leverage the computational resources of common graphics hardware allowing the overall process to be reasonably fast (a few minutes in the data sets of [Strecha et al., 2008] featuring from 8 to 30 images of 6 Mpix, on an Intel Xeon 3.0GHz CPU with an NVIDIA 8800 GTX GPU).

Errors in the multi-view stereo case can come from noise in the images, but as the reconstruction involves matching points in different images, the corresponding 3D error distribution is complex and cannot be modeled as simply as in range scanning case. In addition the noise level is much higher in such passive stereo. Mismatches are also almost inevitable leading to gross outliers. Depending on the geometry of the cameras and the repetitiveness of texture patterns, these mismatches may even aggregates in structured clusters of outliers producing phantom structures in the point cloud. Another limitation of passive stereo is the highly non-uniform density of samples that depends on the amount of texture on the scene and object. While visibility filtering and expansion techniques combining heuristic-based optimizations have been able to improve the quality of point clouds from stereo as in [Furukawa and Ponce, 2007, Goesele et al., 2007], standard point clouds from multi-view such as the two described acquisition methods have notoriously higher levels of noise and ratio of outliers that point clouds acquired with laser range finding.

1.3 Conclusion

In this chapter, techniques to acquire unorganized point clouds sampling a target object or a scene have been described both for range scanning devices and for passive stereo. Despite strong differences in sampling quality, the two family of methods obviously share the property that acquired points are "visible" from one or several sensors or emitters meaning that the corresponding line-of-sights travels through empty space only between the object of interest and the sensor or emitter. This simple information is instrumental in the surface fitting framework presented in the next chapter. Finally, even when this information has been thrown away and cannot be easily inferred, this information may be synthesized in the case of densely and uniformly sampled closed surfaces without outliers: the *hidden point removal operator* of [Katz et al., 2007] may still be applied afterwards to simulate acquisitions from virtual sources, an example is shown in Figure 1.7 (for one source, it only amounts to computing the convex hull of the projection center and a spherically flipped point cloud).

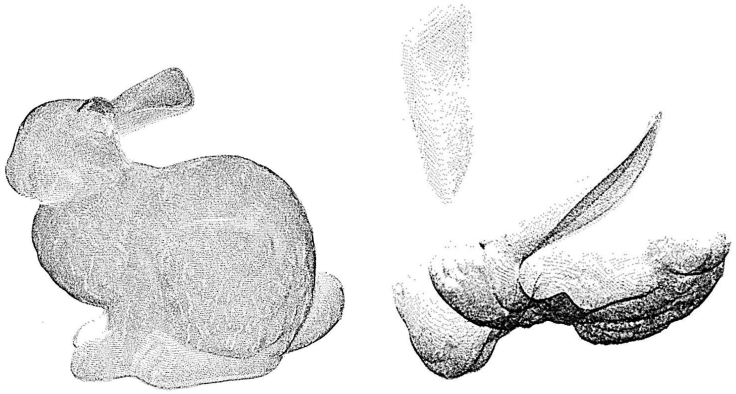


Figure 1.7: ***Applying the hidden point removal operator of [Katz et al., 2007]*** *to the Stanford bunny from a virtual source.*

CHAPTER 2

Surface fitting framework

After an overview of related work on fitting surfaces to sets of points, this chapter presents a general visibility-based surface fitting framework. The problem of surface reconstruction from point clouds is cast as a binary labeling of cells in a cell complex. The cell complex is chosen so as to capture the surface with a subset of its facets while the labeling problem exploits visibility constraints from the line-of-sights associated with the acquired points. A globally optimal labeling and a corresponding surface can efficiently be computed as a minimum s-t cut in a network graph derived from the cell complex.

Contents

2.1 Previous work on fitting surface to point sets

From a set of points $\mathcal{P} = \{P_i\}_{i\in[\![1,N]\!]}$ in $\mathbb{E}^3$ assumed to sample a solid model, the task consists in computing a surface $\mathcal{S}$ from $\mathcal{P}$. The reconstructed surface $\mathcal{S}$ is typically represented as a triangulated mesh directly usable for further processing. This surface $\mathcal{S}$ should match the surface of the original model both geometrically and topologically. In $\mathbb{E}^2$, if an interpolatory curve is sought after and in the absence of noise or outliers not sampling the curve, an optimal curve reconstruction is the polygon that connects the samples the way there are connected along the original curve. In contrast, in $\mathbb{E}^3$, the surface reconstruction problem is in general ill-posed. Depending on the sampled surface and the properties of the sampling, the difficulty of meeting these criteria also varies: the problem is made more challenging by sparsity, varying density, noise, outliers, non-smoothness and the presence of boundaries in the original surface.

As noted in the introduction chapter, this problem of reconstructing a surface from scattered points sampling a physical shape is motivated by numerous applications. Various surface reconstruction algorithms have thus been developed. In this section, we give an overview of the existing techniques. Two major categories of surface reconstruction methods exist: implicit surface methods and Delaunay-based methods. A few other methods based on deformable models or shape template are also discussed.

2.1.1 Implicit methods

The first approach to surface reconstruction constructs a function of space from the samples and defines the surface implicitly as a level-set of the function which permits smooth and approximating surface reconstruction. Other advantages include the fact that the output surface is always the watertight boundary of a solid and that singularities can be avoided. To output a triangulated mesh, these method are often required to apply an isosurface extraction algorithm such as the common Marching Cubes [Lorensen and Cline, 1987] or a more advanced, feature preserving variant [Kobbelt et al., 2001].

Distance functions

The first example of implicit method is [Hoppe et al., 1992] which estimates the tangent plane at each sample point from its k nearest neighbors. A consistent orientation of these planes is found with a minimum spanning tree optimization and the function is defined as the distance to the tangent plane associated to the closest point in space.

In [Curless and Levoy, 1996], the authors acknowledge the importance of the scanning process and exploit line of sight information to blend weighted

distance functions derived from range images. For efficiency reasons, the domain is restricted to a thin shell around the samples and the output of the method is thus a non-watertight reconstruction with boundaries.

Radial basis functions

Later methods represent the function as a weighted sum of basis functions, typically radial basis functions (RBF), more relevant to solve interpolation problems from scattered data [Franke and Nielson, 1980]. The value of the functions is known at constraint points where RBFs are placed. The weights are then globally solved to satisfy the constraints either exactly (for an interpolating reconstruction) or approximately and minimize a smoothness measure. The smoothest function that fits the points is zero everywhere. To avoid this trivial solution, [Carr et al., 2001] has to impose on- and off-surface constraints. In [Ohtake et al., 2004], compactly supported RBFs with adaptive support are relied upon to handle noise.

Partition of unity and moving least-squares

Other recent approaches construct local functions near the sample points and blend them together (using locally supported weight functions that sum to one) to obtain the implicit function. In [Ohtake et al., 2003], the multi-level partition of unity implicit surface representation is introduced: low-degree polynomials approximate the shape of the surface in each cell of an adaptive octree and an efficient implementation is demonstrated to handle large sets of points. Moving least squares (MLS) [Lancaster and Salkauskas, 1981] can naturally handle moderate amount of noise and be used to define similar implicit functions with signed distance to local planes as local approximants, yielding the implicit MLS method of [Shen et al., 2004]: reconstruction guarantees are provided for sufficiently dense and uniform point clouds [Kolluri, 2008]. A related but different method is the projection-based MLS of [Levin, 2003] where the surface is sought as the fixed-point of a parametric fit procedure. [Alexa et al., 2003] introduced the technique to the field of computer graphics with a polynomial fitting step. Numerous variants exists among which [Fleishman et al., 2005] where sample neighborhoods are found with a least-median-of-squares estimator from robust statistics to cope with noise and to preserve sharp features.

Indicator functions

Another choice of function is the indicator valued one inside the object and zero outside. [Kazhdan, 2005, Kazhdan et al., 2006] note that the gradient of the indicator is zero everywhere, except on the surface where it equals the surface normal. A vector field $\overrightarrow{V}$ is computed from the oriented input samples ($\mathcal{P}$ equipped with estimated oriented normals) and the gradient of

the indicator is aligned in the least-squares sense with this vector field by minimizing the functional $\int_\Omega \left\| \overrightarrow{V}(\boldsymbol{x}) - \overrightarrow{\nabla}\chi(\boldsymbol{x}) \right\|^2 \mathrm{d}\boldsymbol{x}$ where the indicator χ is relaxed to take any real value. Applying the calculus of variations leads to a Poisson problem $\Delta\chi = \operatorname{div} \overrightarrow{V}$. Initially solved in the Fourier domain [Kazhdan, 2005], locally supported RBFs are used in [Kazhdan et al., 2006] over an adaptive octree for efficiency and produce excellent results making the method very competitive in practice as the global $\mathcal{L}^2$ minimization naturally brings resilience to noise.

Methods such as [Hornung and Kobbelt, 2006c, Lempitsky and Boykov, 2007], based on the minimal surfaces framework with graph cuts of [Boykov and Kolmogorov, 2003] also belongs to this category: values of the indicator function are assigned to whole elementary volumes over a regular grid so as to globally minimize an energy with a combinatorial minimum s-t cut optimization. Post-processing is required to lessen artifacts from the regular grid discretization.

This family of implicit approaches is sometimes limited by their sensitivity to noise, outliers or non-uniform sampling or even simply by the lack of reliable and consistent normal estimates.

2.1.2 Deformable models

Since the seminal paper [Witkin and Terzopoulos, 1988] on active contours, many applications of deformable models that evolve under the sum of external and internal energies have been found in computer vision and graphics. In general, these evolution methods require a good initialization and are prone to local minima. In the level set framework of [Dervieux and Thomasset, 1979, Osher and Sethian, 1988], the time-varying model is represented as the 0-level set of a function in $\mathbb{R}^3$ whose level sets are all evolved. In [Whitaker, 1998], the evolution guides the model towards a maximum a posteriori by considering the squared error along the line of sights. [Zhao et al., 2001] propose to minimize the functional $\left[\int_{\mathcal{S}} d^p(\boldsymbol{x}, \mathcal{P})\, \mathrm{d}\mathcal{S}\right]^{\frac{1}{p}}$ for $p > 0$, measuring the distance of the surface to the samples. More recently, [Sharf et al., 2006] evolve an explicit deformable mesh in a scalar field guided by the local feature size in a coarse to fine manner to avoid local minima and capture details.

2.1.3 Delaunay methods

The other most common approach to surface reconstruction comes from the computational geometry community and is based on the Voronoi diagram

and its dual, the Delaunay triangulation[1]. Delaunay-based methods follow the initial intuition of [Boissonnat, 1984] of using a Delaunay triangulation for surface reconstruction: the underlying idea is that when the sampling is noise-free and dense enough, points close on the surface should also be close in space. Eliminating facets of Delaunay tetrahedra according to some criteria should then allow the reconstruction of a triangulated mesh.

The α-shapes of [Edelsbrunner and Mücke, 1994] define subsets of Delaunay tetrahedra for each parameter α leading to a discrete family of shapes approximating the point sets at various levels of details.

Among Delaunay-based methods, perhaps the most well-known algorithms are the Crust [Amenta et al., 1998b, 2001] and the Cocone [Amenta et al., 2002, Dey and Goswami, 2003] families of algorithms. Crust algorithms exploit the fact that Voronoi cells of points on the surface are elongated in a direction perpendicular to the inferred surface. The extremal vertices of these cells, called *poles* can be used to estimate the medial axis and filter out facets not belonging to the surface. The Power Crust [Amenta et al., 2001] is an extension, more robust for realistic inputs, that instead relies on the power diagram, a weighted Voronoi diagram of the poles. A simple modification, suggested in [Mederos et al., 2005], improves the robustness of this method to noise. Cocone algorithms use poles in a simpler way to compare facets normal with the vectors to poles. The Robust Cocone [Dey and Goswami, 2006] generalizes the definition of poles to cope with a specific noise model. While [Amenta and Bern, 1999] was the first to provide theoretical guarantees for smooth surfaces with the notion of local feature size and ϵ-sampling, several of the mentioned algorithms are also provably correct in the absence of noise and outliers or under specific noise model related to the local feature size.

In contrast with these computational geometry approaches, [Chaine, 2003] proposes to translate the surface convection scheme of [Zhao et al., 2001] over the Delaunay triangulation of the input points. The ball pivoting algorithm of [Bernardini et al., 1999] also differs from the above approaches. It avoids computing the entire Delaunay triangulation of the input points to then "throw away" most of it but is limited in robustness.

Degradations of the input data may make these local techniques fail. A notable exception to this rule is the spectral surface reconstruction of [Kolluri et al., 2004] which applies a global partitioning strategy (to be discussed in 2.2.2) to label Delaunay tetrahedra as inside or outside the surface and robustly handles quantities of outliers. A more detailed review of Delaunay-based surface reconstruction can be found in the recent survey of [Cazals and Giesen, 2006].

The suggested classification of surface reconstruction algorithms is not

[1]Definitions are given in Appendix B.

rigid. As seen above, a first example is [Chaine, 2003] which formulates the flow of [Zhao et al., 2001] on a Delaunay triangulation. Another is [Boissonnat and Cazals, 2002] where a smooth surface is reconstructed with the concept of natural neighbors defined on the Delaunay triangulation of the samples for improved robustness to non-uniform sampling. In [Samozino et al., 2006], the authors place RBF centers on the medial axis estimated from the Voronoi diagram of the samples. The recent work of [Alliez et al., 2007] mixes together a Delaunay-based approach and an implicit one with a spectral method: a Voronoi diagram is used to estimate a tensor field representing normal direction and confidence, then a generalized eigenvalue problem is solved to compute an implicit function whose gradient is most aligned with the direction given by the tensor field. The method is thus not interpolatory, seems robust to noise. Due to the reliance on Voronoi cells, it is however not robust to outliers. Furthermore, its computational requirements seem too high and may prevent its application to large amounts of data.

It should be noted that most of these works only target reconstruction from high-quality range data. Recently, in the field of multi-view stereo, [Goesele et al., 2007, Furukawa and Ponce, 2007, Bradley et al., 2008] presented specifically designed heuristics to filter, densify and improve the quality of point clouds or depth maps from passive stereo and to make some of the most robust techniques above (namely [Kazhdan et al., 2006] and MLS surface reconstruction) still applicable to these refined point clouds.

In the following section, we provide some background on cell complexes and the use of minimum s-t cuts for optimal surface reconstruction in perspective of the more general cut minimization problem. We also discuss the shortcomings of the previous cut-based reconstruction methods that bear some similarity to our approach.

2.2 Background and notations

2.2.1 Cell complexes and binary labelings of cells

We first give some definitions of typical cell complexes: simplicial complexes and polyhedral complexes.

Definition (Simplex)**.** *A k-simplex or a simplex of dimension k is the convex hull of $k+1$ affinely independent points* i.e. *a set of $k+1$ points $\mathcal{P} = \{P_1, \ldots, P_{k+1}\}$ generating an affine space of dimension k.*

Any subset of $l+1 \leq k+1$ points in $\mathcal{P}$ defines an l-simplex which is a face of the simplex. For instance:

- a 0-simplex is a point, it only has 1 0-face;
- a 1-simplex is a segment, it has 2 0-faces and 1 1-face;
- a 2-simplex is a triangle, it has 3 0-faces, 3 1-faces and 1 2-face;

- a 3-simplex is a tetrahedron, it has 4 0-faces, 6 1-faces, 4 2-faces and 1 3-face,
- etc...

Definition (Simplicial complex)**.** *A simplicial complex $\mathcal{C}$ is a finite set of simplices that satisfies the following two properties:*

1. *any face of a simplex in $\mathcal{C}$ is also a simplex in $\mathcal{C}$, and*
2. *two simplices in $\mathcal{C}$ either do not intersect, or their intersection is a simplex of smaller dimension which is their common face of maximal dimension.*

Common examples of simplicial complexes are triangulations in 2D or tetrahedrizations in 3D. The notion of simplex and simplicial complex is generalized with polytopes and polyhedral complexes.

Definition (Polytope)**.** *A polytope is the convex hull of a finite set of points $\mathcal{P}$ in $\mathbb{E}^d$.*

A polytope is equivalently the intersection, if it is bounded, of a finite number of half-spaces. A hyperplane supporting a polytope is a hyperplane having a non-empty intersection with the polytope and such that the polytope (or $\mathcal{P}$) is completely contained in one of the half-spaces induced by the hyperplane. The definition of the faces of a polytope is more elaborate than for a simplex: a face is the intersection of a polytope with a supporting hyperplane. A face is the convex hull of a subset of the points $\mathcal{P}$ defining the polytope. A polytope of dimension[2] k is also called a k-polytope.

Similarly to a simplicial complex, a polyhedral complex[3] can be defined as follows:

Definition (Polyhedral complex)**.** *A polyhedral complex $\mathcal{C}$ is a finite set of polytopes that satisfies the following two properties:*

1. *any face of a polytope in $\mathcal{C}$ is also a polytope in $\mathcal{C}$, and*
2. *two polytopes in $\mathcal{C}$ either do not intersect, or their intersection is a polytope of smaller dimension which is their common face of maximal dimension.*

Simplices and simplicial complexes are only a special case of respectively polytopes and polyhedral complexes. The polytopes that constitute a polyhedral complex are called the faces of the complex. A face of dimension k is called a k-face. 0-faces are called the vertices, 1-faces are called the edges and in dimension d, $d-1$-faces are called facets and d -faces are called cells.

[2]The dimension of a convex set is defined as the dimension of its affine hull (the smallest affine set containing it).

[3]Sometimes called *polytopial complex* in the literature.

Complexes can be further generalized with CW-complex[4] in algebraic topology. Since we are only interested in geometric properties related to CW-complexes, a simplified definition of *cell complexes* will be given, consistent with our needs. For a more general and rigorous treatment of CW-complexes in topological spaces and algebraic topology, we refer the reader to [Hatcher, 2002].

Definition (Closed cell)**.** *A closed k-cell or a closed cell of dimension k is a closed topological k-ball in $\mathbb{E}^d$,* i.e. *, a subset of $\mathbb{E}^d$ which is homeomorphic to a closed Euclidean k-ball.*

Definition (Open cell)**.** *An open k-cell or an open cell of dimension k is an open topological k-ball in $\mathbb{E}^d$,* i.e. *, a subset of $\mathbb{E}^d$ which is homeomorphic to an open Euclidean k-ball.*

A k-polytope (and thus a k-simplex) is a closed k-cell. An open or closed 0-cell is a point in $\mathbb{E}^d$.

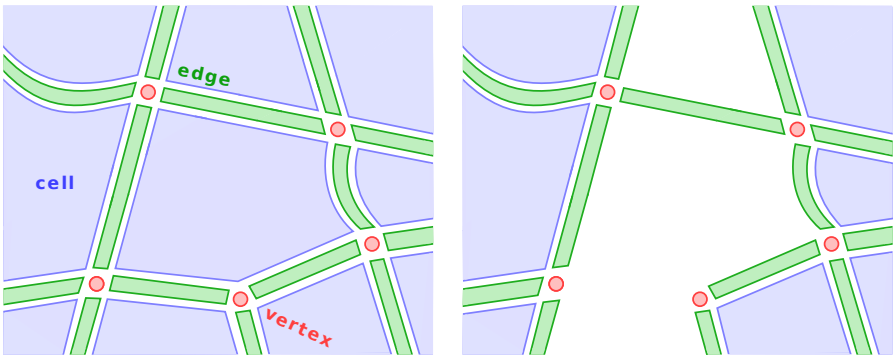
Definition (Cell complex)**.** *A cell complex is a partition of a set $U \subseteq \mathbb{E}^d$ into open cells such that for each open k-cell C in the partition of U, there exists a homeomorphism h from the closed k-ball into U which satisfies the following two properties:*

1. *the restriction of h to the interior of the closed ball is a homeomorphism onto the cell C,*
2. *the image under h of the boundary of the closed ball is a finite union of elements of the partition whose cell dimension is less than k (called faces of C).*

Polyhedral complexes (and obviously simplicial complexes) match this definition but the faces defined above are actually closed k-cells in the cell complex terminology, and that the term cell above designates closed k-cells of maximum cell dimension here. We will use the above definitions of faces, vertices, edges, facets and cells throughout this thesis.

A complex $\mathcal{C}$ is a k-complex or a complex of dimension k if the maximal (cell-)dimension of the faces in $\mathcal{C}$ is exactly k. We denote by $\mathcal{C}_l$ the set of l-faces of a k-complex $\mathcal{C}$ ($0 \leq l \leq k$). A k-complex is *homogeneous* or *pure* if and only if any face of $\mathcal{C}$ is a face of some closed k-cell in $\mathcal{C}$. The *domain* of a complex $\mathcal{C}$ is denoted by $\mathrm{dom}(\mathcal{C})$ and is the region of $\mathbb{E}^d$ formed by the points in $\mathbb{E}^d$ belonging to the faces of $\mathcal{C}$. A k-complex is connected if its domain is connected. In this thesis, only pure connected cell complexes are considered whose domain is $\mathbb{E}^3$ by including infinite cells. In Chapters 3 and 4, the considered cell complex is a particular simplicial complex, a triangulation of the input points, the Delaunay triangulation. In Chapter 7, the cell complex is the polyhedral complex induced by a binary partitioning of space with

[4] CW stands for closure-finite weak topology.



(a) A complex (b) A connected but impure complex

Figure 2.1: ***Examples of cell complexes.***

planes and in Chapter 8, the cell complex is induced by a generalized binary partitioning of space with oriented surfaces.

Two faces of a complex are incident if one is contained in the other and their dimension differ by one. Two cells (*i.e.* , k-cells in a k-complex) are adjacent if they share a common incident facet (a $k-1$-face). Since cells of a complex only intersect in a finite number of facets and as we consider cell complexes with a finite number of cells, the finite (undirected) graph of cell adjacency in a complex is defined as follows:

Definition (Cell adjacency graph)**.** *The cell adjacency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of a cell complex $\mathcal{C}$ is a graph with n vertices $\mathcal{V} = \{v_1, \ldots, v_n\}$, in one-to-one correspondence with the n cells of the complex. Each edge e in $\mathcal{E}$ corresponds to a facet F of $\mathcal{C}$,* i.e. *, the intersection of two adjacent cells C_i and C_j.*

Binary values can be assigned to cells with binary labelings of cells:

Definition (Cell binary labeling)**.** *A cell binary labeling l of a k-complex $\mathcal{C}$ is a map from $\mathcal{C}_k$ to $\{0, 1\}$.*

In the sequel, $l(C)$ will be shortened as l_C. A notion of orientation for facets of a cell complex is now introduced:

Definition (Oriented facet)**.** *Let F be a facet of a k-complex incident to the two adjacent cells C_i and C_j* (i.e. $C_i \cap C_j = F$)*, $F^{C_i \to C_j}$ denotes the oriented facet oriented from C_i to C_j.*

The directed cell adjacency graph of a cell complex $\mathcal{C}$ can also be defined with an edge corresponding to each oriented facet in the complex $\mathcal{C}$. At last, the definition of a pseudo-surface associated to a binary labeling can be given:

Definition (Pseudo-surface)**.** *A pseudo-surface $\mathcal{S}$ in a k-complex $\mathcal{C}$ induced by a binary labeling l of cells is the union of all the oriented facets from* 1*-labeled cells to* 0*-labeled cells:*

$$\mathcal{S} = \bigcup_{\substack{C_i, C_j \in \mathcal{C}_k \\ l_{C_i}=1 \\ l_{C_j}=0 \\ C_i \cap C_j = F}} F^{C_i \to C_j} \ .$$

A pseudo-surface can also be interpreted as an oriented $k-1$-complex. More interestingly, with the convention that $l_C = 0$ means that the cell C is outside or empty and that $l_C = 1$ means that the cell C is considered inside, labeling can describe indicator functions: there is a one-to-one correspondence between binary labelings of a cell complex and the union of oriented facets bounding a solid formed by inside cells in 3D.

Having defined what a cell complex is, and how some particular unions of oriented facets can be interpreted as oriented surfaces, the next section proposes to cast the problem of finding optimal (oriented) surfaces bounding solids as finding an optimal cut on the directed cell adjacency graph of a given cell complex.

2.2.2 Surface reconstruction with minimum cuts

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a *directed* graph with vertices $\mathcal{V} = \{v_1, \ldots, v_n\}$ and oriented edges $\mathcal{E}$ with weights w_{ij}. Graph partitioning consists in removing the edges connecting two sets of vertices, that is finding two disjoint sets $\mathcal{V}_\mathcal{S}$ and $\mathcal{V}_\mathcal{T}$ such that $\mathcal{V}_\mathcal{S} \cup \mathcal{V}_\mathcal{T} = \mathcal{V}$, $\mathcal{V}_\mathcal{S} \cap \mathcal{V}_\mathcal{T} = \emptyset$. This partition $(\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T})$ is called a *cut* and is assigned a *cost*: the sum of the weights of the edges going from $\mathcal{V}_\mathcal{S}$ to $\mathcal{V}_\mathcal{T}$ (the oriented edges "crossed" by the cut):

$$\mathrm{cost}(\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T}) = \sum_{\substack{v_i \in \mathcal{V}_\mathcal{S} \\ v_j \in \mathcal{V}_\mathcal{T}}} w_{ij} \ .$$

This cost can be seen as a measure of similarity between the two sets $\mathcal{V}_\mathcal{S}$ and $\mathcal{V}_\mathcal{T}$. If we consider the directed cells adjacency graph of a cell complex, each vertex is a cell and each edge is an oriented facet, and partitioning a weighted version of this graph is equivalent to assigning binary labels to cells according to oriented facets weights. A binary labeling of cells corresponds to a graph partitioning and once weights have been assigned to oriented facets, the cost of a cell binary labeling can be defined. In the sequel, we assume that the label 0 for a cell C corresponds to the fact that its vertex v_C belongs to $\mathcal{V}_\mathcal{S}$, and similarly for the label 1 and $\mathcal{V}_\mathcal{T}$: in other words, a labeling l is the indicator function of $\mathcal{V}_\mathcal{T}$.

Since the total number of possible cuts is 2^n, a combination of efficient algorithms and restricted sets of cuts are needed to find a minimum cut.

Spectral partitioning

Spectral clustering is a first class of methods for computing optimal partitions of graph. Given an *undirected* similarity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with non-negative weights ($w_{ij} = w_{ji} \geq 0$), we define:

- $d_i = \sum_j w_{ij}$ and $D = (d_i)^T \in \mathbb{R}^n$,
- l_i, a relaxed label assigned to the graph vertex v_i and $l = (l_i)^T \in \mathbb{R}^n$,
- $W = (w_{ij}) \in \mathcal{M}_{n,n}(\mathbb{R})$

Applications of spectral partitioning make use of graphs induced by the symmetrized k-nearest neighbors, ϵ-neighbors or even fully connected graphs. In our context, the undirected cells adjacency graph of the complex is an obvious candidate. If we note $L = D - W$, the symmetric and positive semi-definite *unnormalized Laplacian matrix* of the graph, then we have (other properties of this matrix can be found in [Mohar, 1991, 1997]):

$$l^T L\, l = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(l_i - l_j\right)^2 \ .$$

If l_i takes only binary values then this is (almost) exactly the cost of the corresponding cut in the graph and efficient algorithms exist to compute the minimum cut [Stoer and Wagner, 1997] (requiring $\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T} \neq \emptyset$). With a relaxed labeling and also avoiding the trivial solution, $l^T L\, l$ is minimized when l is the second eigenvector of L (the first eigenvalue of L is 0). From this relaxed labeling (for instance computed with Lanczos algorithm to find the first eigenvectors of the sparse matrix L), a binary labeling of vertices can then be found with the k-means algorithm [Steinhaus, 1956] or thresholding. However, the defined cost does nothing to prevent small sets of nodes from being isolated. A more useful spectral partitioning method [Shi and Malik, 2000] considers the following normalized cost:

$$\text{cost}(\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T}) = \sum_{\substack{v_i \in \mathcal{V}_\mathcal{S} \\ v_j \in \mathcal{V}_\mathcal{T}}} w_{ij} \Bigg/ \sum_{\substack{v_i \in \mathcal{V}_\mathcal{S} \\ v_j \in \mathcal{V}}} w_{ij} + \sum_{\substack{v_i \in \mathcal{V}_\mathcal{S} \\ v_j \in \mathcal{V}_\mathcal{T}}} w_{ij} \Bigg/ \sum_{\substack{v_i \in \mathcal{V}_\mathcal{T} \\ v_j \in \mathcal{V}}} w_{ij} \ ,$$

which penalizes small clusters. Unfortunately, minimizing exactly this balanced cost (or finding a corresponding binary labeling) is an NP-complete problem. If l_i is relaxed, it can be shown that minimizing this new cost leads to minimizing:

$$l^T \tilde{L} l = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{l_i}{\sqrt{d_i}} - \frac{l_j}{\sqrt{d_j}} \right)^2 \ ,$$

where $\tilde{L}$ is a normalized[5] Laplacian matrix $\tilde{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$ (a standard reference for normalized graph Laplacians is

[5] $0^{-1/2}$ is replaced by 0.

[Chung, 1997]). As above, computing the second eigenvector of $\tilde{L}$ allows to infer a binary labeling of vertices with the k-means algorithm [Steinhaus, 1956] or appropriate thresholding.

Surface reconstruction from point clouds has already been expressed as a problem equivalent to computing a normalized cut. The spectral surface reconstruction of [Kolluri et al., 2004] applies this criterion to partition subsets of Delaunay tetrahedra. The method however varies on the presented scheme by modifying the eigenvalue problem: the normalized Laplacian matrix $\tilde{L}$ involved in the objective function is altered by allowing negative weights and modifying its diagonal to make the matrix positive definite. According to the authors, these adjustments in practice greatly improve robustness and increase speed, but they also void the interpretation of the solution as an optimal normalized cut. The method is nevertheless still quite slow, could not be applied to large data sets and besides requiring two successive partitioning steps, several additional ad hoc treatments seem to be needed for it to be applied to real data (filtering out spurious tetrahedra by thresholding and restricting the labeling to tetrahedra near the sample points along their line of sights).

Minimum s-t cuts

Another approach to graph partitioning adds two special "terminal" vertices to $\mathcal{V}$, the source s and the sink t. The weights are restricted to non-negative values but asymmetry is now allowed. In addition to edges to its incident vertices, each vertex v_i now has links to s and t respectively weighted s_i and t_i. An s-t-cut $(\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T})$ is a cut such that $s \in \mathcal{V}_\mathcal{S}$ and $t \in \mathcal{V}_\mathcal{T}$. The cost of such a cut may be split as follows:

$$\text{cost}(\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T}) = \sum_{\substack{v_i \in \mathcal{V}_\mathcal{S} \setminus \{s\} \\ v_j \in \mathcal{V}_\mathcal{T} \setminus \{t\}}} w_{ij} + \sum_{v_i \in \mathcal{V}_\mathcal{S} \setminus \{s\}} t_i + \sum_{v_i \in \mathcal{V}_\mathcal{T} \setminus \{t\}} s_i \ .$$

This cost can be interpreted as an energy $\text{E}(\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T})$ attached to the corresponding partition with a "regularizing" term between the $\mathcal{V}_\mathcal{S}$ and $\mathcal{V}_\mathcal{T}$ sets (the sum of edge weights w_{ij} which is actually the cost of the cut without considering the terminals) and a "data" term for $\mathcal{V}_\mathcal{S}$ and $\mathcal{V}_\mathcal{T}$ (the sums of the link weights s_i and t_i). The minimum s-t cut problem consists in finding an s-t cut $(\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T})$ with the smallest cost. According to the Ford-Fulkerson theorem [Ford and Fulkerson, 1962], this problem is the same as computing the maximum flow from the source s to the sink t: several efficient algorithms with low-polynomial complexity have been developed to solve this problem, making it possible to *globally* minimize the energy $\text{E}(\mathcal{V}_\mathcal{S}, \mathcal{V}_\mathcal{T})$. The two classes of such algorithms are variants of the original augmenting paths [Dinic, 1970] and push-relabel algorithms [Goldberg and Tarjan, 1988] (both

of $\mathcal{O}\left(|\mathcal{V}|^2\,|\mathcal{E}|\right)$ complexity). Detailed descriptions of these algorithms in combinatorial optimization can be found in [Tarjan, 1983, Cook et al., 1998].

By building an appropriate graph, many segmentation problems in computer vision or graphics can be thus solved globally and efficiently [Greig et al., 1989]. More generally, global minimization of specific classes of energy is achievable with minimum s-t cuts [Kolmogorov and Zabih, 2004, Freedman and Drineas, 2005]. Most often, methods using minimum s-t cuts for optimal binary segmentation or partitioning for curve or surface reconstruction straightforwardly apply the minimal surfaces with *graph cuts* framework of [Boykov and Kolmogorov, 2003] by slicing the whole domain of interest with a regular grid and interpreting the previous equation as the sum of the discretizations of an integral over the interface (the surface $\mathcal{S}$) between $\mathcal{V}_\mathcal{S}$ and $\mathcal{V}_\mathcal{T}$ and two integrals over $\mathcal{V}_\mathcal{S}$ (the outside volume V_{out}) and $\mathcal{V}_\mathcal{T}$ (the inside volume V_{in}):

$$\mathrm{E}(\mathcal{S}) = \int_{\mathcal{S}} f(\boldsymbol{x})\,\mathrm{d}\mathcal{S} + \int_{V_{\text{out}}} g_{\text{out}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_{V_{\text{in}}} g_{\text{in}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\ .$$

Often employed over these regular grid graphs, a popular augmenting paths algorithm is the one of [Boykov and Kolmogorov, 2004]. Although its theoretical complexity $\mathcal{O}\left(|\mathcal{V}|^2\,|\mathcal{E}|\,\mathrm{cost}(\mathcal{V}^\star_\mathcal{S}, \mathcal{V}^\star_\mathcal{T})\right)$ is worse than standard algorithms, it is preferred due to better empirical performance.

Methods that rely on minimum s-t cuts for optimal surface reconstruction either directly from images or from point clouds adopt this point of view which has several weaknesses. First, a regular subdivision of space seriously impedes the scalability of minimum s-t cuts as empty space has to be modeled explicitly. It also requires extensive post-processing to remove the stair-cases artifacts on the extracted surface (a continuous version of these minimal s-t cut surfaces was proposed in [Appleton and Talbot, 2006]). Then, the area-based regularization term is the cause of the "shrinking bias"[6]: the optimal surface for an energy reduced to this term only is the trivial empty one. Minimizing an energy including such a term coupled with other terms often results in over-smoothing. Workarounds include restricting the domain of interest or adding a uniform balloon force which requires data-specific adjustment and is unable to recover thin protusions and concavities. In the field of surface reconstruction from point clouds, the approach of [Hornung and Kobbelt, 2006c] unfortunately suffers from both problems: the domain is regularly subdivided with a grid that introduces metrication errors and requires a special post-processing step to smooth out grid artifacts. The computational burden of this grid is limited thanks to the use of "banded" graph cuts of [Lombaert et al., 2005] which actually reduce the minimum s-t cuts optimization to a local optimization in a neighborhood

[6] Typical of numerous regularizations techniques but exacerbated by the global optimum property of minimum s-t cuts.

of the initially computed proxy surface. In [Lempitsky and Boykov, 2007], the flux of a coarsely oriented vector field is maximized. However, to make this approach robust against undersampling and outliers, the authors resort to an area term and a regional term based on lines of sight (a simple non-uniform balloon force). Also, to lower the computational cost, the authors use a dedicated heuristic that requires proper initialisation to speed up the maximum flow computation on the voxelized volume.

Alternatives to regular grids exist: [Kirsanov and Gortler, 2004] first proposed to use graph cuts on cell complexes to globally optimize surface functionals and developed the idea of using random sparse complexes for their flexibility over regular subdivisions. In surface reconstruction from images, [Vogiatzis et al., 2007, Sinha et al., 2007] propose multi-resolution regular subdivisions. In [Labatut et al., 2007], we propose a sparse subdivision adaptive to sample measures with the Delaunay triangulation.

Our general surface fitting method circumvents the common drawbacks of graph cuts for surface reconstruction from images or point clouds. Instead of imposing a regular grid, we exploit the adaptivity of a cell complex guided by the input data. Moreover, a visibility term taking into account the acquisition procedure is proposed: it explicitly avoids the empty surface solution. In contrast with methods purely based on the minimal surfaces of [Boykov and Kolmogorov, 2003], our optimization problem is purely discrete and does not approximate a continuous one.

2.3 Overview of our approach

All surface reconstruction algorithms presented in the next chapters follow the same overall scheme:

Algorithm 1 Summary of our approach

1: Compute a set of points $\mathcal{P}$ with associated visibility sets v,
2: Compute a cell complex from this set of points,
3: Compute an optimal binary labeling $l^\star = \arg\min_l \mathrm{E}(l)$ of its cells *w.r.t.* some energy accounting for visibility constraints (plus possibly some others terms),
4: Extract the corresponding oriented pseudo-surface $\mathcal{S}$ from the optimal inside/outside cells binary labeling $l^\star$.

The underlying assumptions are that the sought surface can be approximated with a union of facets from the cell complex and that the sample points sample these facets densely enough.

2.3.1 Points with visibility sets

As noted in the previous chapter, all the considered acquisition methods suppose clear lines of sight between their respective emitter or sensor and the reconstructed point crossing only empty space. In the rest of the thesis we denote by $\mathcal{P} = \{P_1, \ldots, P_N\}$ the finite set of points in $\mathbb{E}^d$ acquired using one of the techniques discussed in Chapter 1. We denote by v a map from $\mathcal{P}$ to a finite set of points in $\mathbb{E}^3$ which associates to each point $P \in \mathcal{P}$ a "visibility set", *i.e.* , a finite set of locations $v(P) = \left\{V_1, \ldots, V_{|v(P)|}\right\}$ such that for all $i \in [\![1, |v(P)|]\!]$, the line segment $[V_i P]$ is a line-of-sight. Depending on the projection, the point V_i may even be rejected towards infinity and located in an infinite cell of the cell complex. We will note $v(P)$ as v_P in the sequel.

2.3.2 Cell complexes from points

The cell complexes considered in the rest of this thesis are all "data-driven": their construction is guided by the initial point cloud and a subset of the facets of the complex is assumed to capture the surface to be reconstructed. The cell complex is either directly computed from the set of points as in Chapters 3 and 4 where the Delaunay triangulation of the points is used, or as in Chapter 6, it requires a pre-processing of the point cloud to extract appropriate structures (see Chapters 7 and 8). In all cases, this reliance on sparse adaptive point-driven complexes enables our surface reconstruction framework to naturally scale to large outdoor scenes, a problematic case for dense graph cut methods over regular subdivision of space. Hence one drawback is that the actual computation of the cell complex is more labor-intensive than a simple regular grid. This is the compromise to pay for greater flexibility. The cell adjacency graph is however typically much sparser and the ensuing optimization faster. An advantage of the facts that the complexes span the whole ambient space and that the optimization relies on visibility, is the added ability to handle closed scenes but also some open scenes: the extracted pseudo-surface can include infinite facets of the complex, allowing pseudo-surfaces to be extracted with a boundary lying on the finite edges of the infinite cells of the complex. Most other surface reconstruction algorithms can only output closed reconstructions.

2.3.3 Optimal cell binary labeling

A pseudo-surface $\mathcal{S}^*$ is sought after so as to minimize visibility constraints imposed by the line-of-sight of the acquired points:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathrm{E}_{\mathrm{vis}}(\mathcal{S}, \mathcal{P}, v) \ .$$

Surface visibility

A surface should never cross the empty space traversed by the various lines of sight attached to the points. Ideally, one would like to minimize the number of conflicts of the lines of sight with the pseudo-surface $\mathcal{S}$ induced by the cell labeling l. This corresponds to the following energy:

$$\mathrm{E}_{\mathrm{vis}}(\mathcal{S},\mathcal{P},v) = \mathrm{E}_{\mathrm{vis}}(l,\mathcal{P},v) = \sum_{P\in\mathcal{P}} \sum_{Q\in v_P} V_{\mathrm{conflict}}\left(l_{C_1^{Q\to P}},\ldots,l_{C_{N_{[QP]}}^{Q\to P}}\right) ,$$

where $C_1^{Q\to P},\ldots,C_{N_{[QP]}}^{Q\to P}$ is the ordered sequence of the $N_{[QP]}$ cells crossed from the emitter/sensor position Q to the point P. Since P is not assumed to lie exactly on a facet of the complex, the sequence is terminated at the facet of complex closest to P as shown in the upper part of Figure 2.2. $N_{[QP]}$ depends on P, Q and the complex. Each oriented facet F ($C_i^{Q\to P}\cap C_{i+1}^{Q\to P}$ for $i\in[\![1,N_{[QP]}-1]\!]$ is intersected by the line segment $[QP]$. V_{conflict} is given by:

$$V_{\mathrm{conflict}}\left(l_{C_1^{Q\to P}},\ldots,l_{C_{N_{[QP]}}^{Q\to P}}\right) = \alpha_{\mathrm{vis}}\,\mathbb{1}\left[\exists i\in[\![1,N_{[QP]}]\!] \quad l_{C_i^{Q\to P}}=1\right] ,$$

where α_{vis} is a constant *w.r.t.* the labeling but can possibly depend on the considered point or line of sight: it is a confidence measure of the point or line of sight. For instance,

1. for dense range images, α_{vis} can be linked to the estimated normal from the range image or to the closeness of the sample point to boundaries of the range as in [Turk and Levoy, 1994, Curless and Levoy, 1996],
2. for passive point clouds from multiple images, α_{vis} can be linked to the photo-consistency score (it could also be linked to the condition number of the triangulation optimization).

Since the trivial labeling $l^0 : c\in\mathcal{C}\to 0$ marking all cells as outside and to which corresponds an empty pseudo-surface satisfies these constraints, the facts that the point is assumed to lie near the surface and that the sensor has to be outside have to be considered. We denote by $C_1^{Q\to P},\ldots,C_{N_{[QP]}}^{Q\to P}$ the same ordered sequence of the $N_{[QP]}$ cells crossed from the ending point position Q to the point P (up to the closest facet to P). We denote by $C_{N_{[QP]}+1}^{Q\to P}$ the cell behind this closest facet to the point P in the direction of the line of sight, *i.e.* each oriented facet $F=\left(C_i^{Q\to P},C_{i+1}^{Q\to P}\right)$ for $i\in$

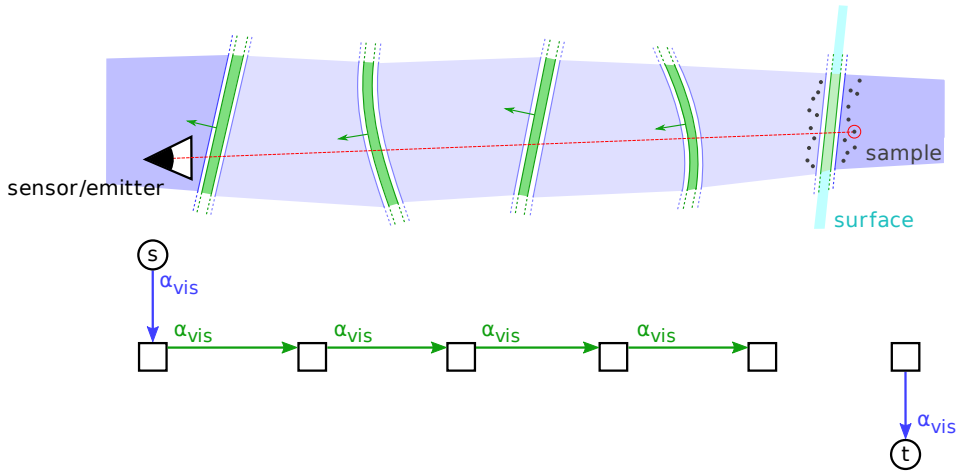


Figure 2.2: ***Visibility and graph construction.*** *A line of sight from a sample point traverses a sequence of cells: the traversed sequence goes from the left darker blue cell to the rightmost light blue cell and the darker blue rightmost cell is the cell "behind" the sample; the darker green facets are considered crossed by the line of sight. To such sequence corresponds the shown graph construction and the assignment of weights to the cells and oriented facets.*

$[\![1, N_{[QP]}]\!]$ is intersected by the line $[QP]$. We define $\mathrm{E}_{\mathrm{vis}}$ as:

$$\begin{aligned}\mathrm{E}_{\mathrm{vis}}(\mathcal{S},\mathcal{P},v) = \mathrm{E}_{\mathrm{vis}}(l,\mathcal{P},v) = \sum_{P\in\mathcal{P}}\sum_{Q\in v_P} & D_{\mathrm{out}}\left(l_{C_1^{Q\to P}}\right) \\ & + V_{\mathrm{conflict}}\left(l_{C_1^{Q\to P}},\ldots,l_{C_{N_{[QP]}}^{Q\to P}}\right) \\ & + D_{\mathrm{in}}\left(l_{C_{N_{[QP]}+1}^{Q\to P}}\right) ,\end{aligned}$$

where:

$$\begin{aligned} D_{\mathrm{out}}\left(l_C\right) &= \alpha_{\mathrm{vis}}\,\mathbb{1}\left[l_C = 1\right] , \\ D_{\mathrm{in}}\left(l_C\right) &= \alpha_{\mathrm{vis}}\,\mathbb{1}\left[l_C = 0\right] . \end{aligned}$$

As announced, a labeling that marks the cell of the emitter/sensor as inside is penalized and a labeling marking the cell behind the point as outside is also penalized in addition to labelings creating conflicts with lines of sight. A major drawback however, is that the subterm V_{conflict} involves a varying number of cells (typically greater than two) and an optimal labeling can thus not be found using a minimum s-t cut on the directed cells adjacency graph of the complex. Instead of penalizing a "conflict" with each line of sight, the

number of misalignments of the surface may be penalized. To this end, $\mathrm{E}_{\mathrm{vis}}$ is replaced with the following new expression:

$$\begin{aligned}\mathrm{E}_{\mathrm{vis}}(\mathcal{S},\mathcal{P},v) = \mathrm{E}_{\mathrm{vis}}(l,\mathcal{P},v) = \sum_{P\in\mathcal{P}}\sum_{Q\in v_P} & D_{\mathrm{out}}\left(l_{C_1^{Q\to P}}\right) \\ & + \sum_{i=1}^{N_{[QP]}-1} V_{\mathrm{align}}\left(l_{C_i^{Q\to P}}, l_{C_{i+1}^{Q\to P}}\right) \\ & + D_{\mathrm{in}}\left(l_{C_{N_{[QP]}+1}^{Q\to P}}\right) ,\end{aligned}$$

where V_{align} is a simple pair-wise subterm defined for two adjacent cells of the complex (since in the above equation the cells are crossed in that order, they are adjacent to each other):

$$V_{\mathrm{align}}\left(l_{C_i}, l_{C_j}\right) \quad = \quad \alpha_{\mathrm{vis}}\, \mathbb{1}\left[\, l_{C_i} = 0 \wedge l_{C_j} = 1 \,\right] .$$

The benefit of this new visibility term is that a globally optimal labeling can be efficiently be computed with minimum s-t cut. The corresponding weight construction is shown in Figure 2.2: the s-link of the vertex representing the cell C_1 is assigned α_{vis}, the t-link of vertex representing the cell $C_{N_{[QP]}}$ behind the closest facet to P is assigned α_{vis} and each oriented facet crossed by the line of sight from P to Q is also assigned α_{vis}. These weight assignments are accumulated over all lines of sight, and computing a minimum s-t on this graph yields a globally optimum labeling.

One might wonder if alternatives would not be better suited to this problem. For instance, simply penalizing crossed cells with a data term, *i.e.* , using the D_{out} subterm for all crossed cells leads to a guided ballooning force (this is one of the additional terms used in [Lempitsky and Boykov, 2007]), known to be not as robust and often requiring data-specific parameter tuning. Making the subterm V_{align} symmetric reduces to a standard regularizing term along each line of sight. The presented visibility term has been designed to avoid these pitfalls and an extensive validation in the next chapters confirms this. Its integral nature, by accumulating weights over all available line-of-sights, explains its robustness to outliers present both in the empty space crossed by the line-of-sights and behind densely sampled surface areas.

Other terms

We have purposely not detailed the construction of the cell complex since it is problem specific. Depending on the sampling density of the points and their line of sights *w.r.t.* the facets of the complex, some additional term might

be needed to compute not only a surface that is approximately visibility-consistent but also a surface avoiding some mislabeling due to scarce visibility information. These additional terms may be a surface quality measure for Delaunay triangulations in Chapters 3 and 4 or the photo-consistency of the surface for the multi-view stereo problem.

2.3.4 Pseudo-surface extraction

As noted in Section 2.2, there is a one-to-one correspondence between binary labelings of cells and inside/outside partitions by pseudo-surfaces. From an optimal cell labeling $l^\star$, a corresponding optimal pseudo-surface $\mathcal{S}^\star$ can be extracted as the interface between 0-labeled cells (outside cells) and 1-labeled cells (inside cells):

$$\mathcal{S}^\star = \bigcup_{\substack{C_i, C_j \in \mathcal{C}_3 \\ l^\star_{C_i}=1 \\ l^\star_{C_j}=0 \\ C_i \cap C_j = F}} F^{C_i \to C_j} \ ,$$

where $\mathcal{C}_3$ is the set of cells of the complex. In the various complexes used in Chapters 3, 4, 7 and 8, the output pseudo-surface is always a polygonal mesh, with either purely triangular facets, more general polygonal facets in Chapter 7 or whole surface patches approximated by triangular meshes in Chapter 8. This output mesh may not be 2-manifold, *i.e.* , it may contain singular vertices or edges, but, if required, it can be converted to a 2-manifold mesh using standard mesh processing techniques such as [Guéziec et al., 2001].

Note that while the labeling of cells is guaranteed to be globally optimal, the pseudo-surface is only optimal *w.r.t.* to the underlying complex. The choice and construction of the complex is a critical part of our approach for the final pseudo-surface S to accurately model the acquired object or scene. To illustrate this and the use of the presented visibility term over different point-driven cell complexes, Figures 2.3 and 2.4 compare reconstructions from noise-free point clouds with the same visibility term and a simple uniform and symmetric term V_{comp} per facet, which penalizes the number of facets in the final pseudo-surface:

$$V_{\text{comp}}\left(l_{C_i}, l_{C_j}\right) = V_{\text{comp}}\left(l_{C_j}, \mathit{Labeling}_{C_i}\right) = \mathbb{1}\left[\, l_{C_i} \neq l_{C_j} \,\right] \ ,$$

for two adjacent cells C_i and C_j such that $C_i \cap C_j = F$. In the minimum s-t cut optimization framework, this term amounts to adding a weight to all oriented facets. The reconstructions were computed on an adaptive region-based octree [Finkel and Bentley, 1974] (at maximum depth 9, with at most one point per octree cell) and on the Delaunay triangulation of the input

points. The input point cloud samples a ground truth model and the visibility sets were obtained with occlusion computation using this ground truth model. In these noise-free examples, the choice of a Delaunay triangulation completely removes the stair-case artifacts typical of graph cuts method over regular subdivisions and obviates any post-processing.

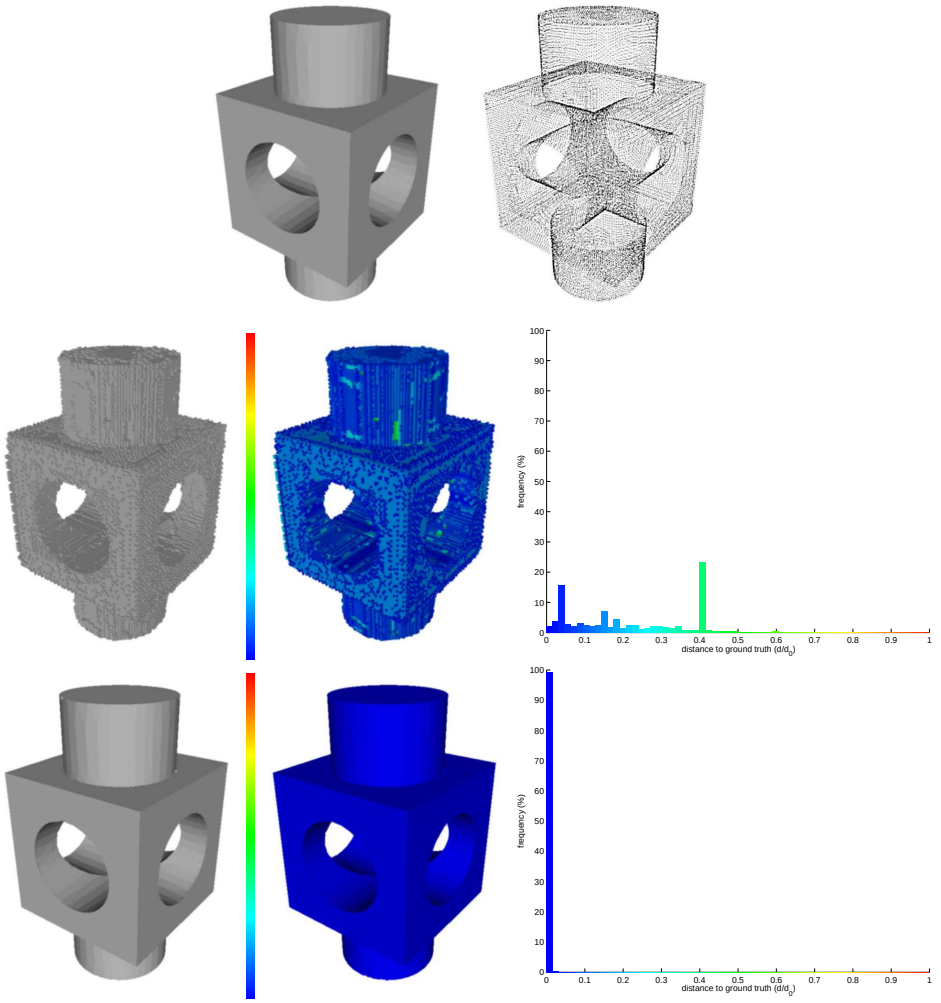


Figure 2.3: ***block data set:*** *comparison of visibility-based reconstructions over different data-driven complexes. First row: ground truth model and input point cloud, second row: reconstruction over an adaptive region-based octree and third row: reconstruction over a Delaunay triangulation. The graphs on the right are histograms of the distance to ground truth of the reconstructions.*

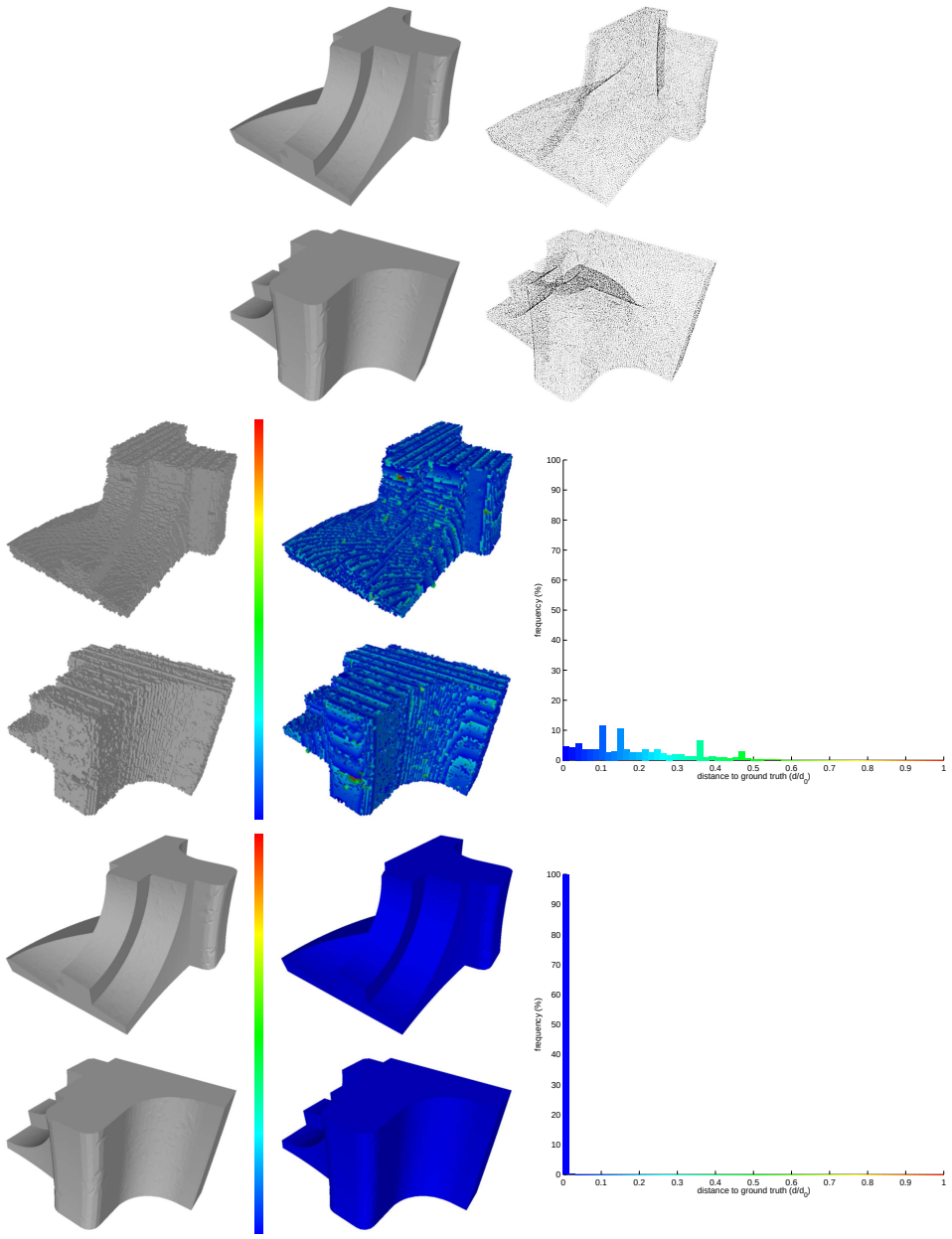


Figure 2.4: ***fandisk data set:*** *comparison of visibility-based reconstructions over different data-driven complexes. First and second rows: ground truth model and input point cloud. Third and fourth rows: reconstruction over an adaptive region-based octree. Fifth and sixth rows: reconstruction over a Delaunay triangulation. The graphs on the right are histograms of the distance to ground truth of the reconstructions.*

2.4 Conclusion

This chapter has introduced our general surface fitting framework. The problem of surface reconstruction from acquired point clouds is cast as the reconstruction of a visibility-consistent pseudo-surface in a data-driven cell complex. This problem reduces to a binary labeling of cells that can efficiently be computed with a minimum s-t cut. Problem-specific cell complexes are used in the subsequent chapters for different surface reconstruction applications.

Part I

Visibility-based surface reconstruction

CHAPTER 3

Surface reconstruction from range data

A first variant of our visibility-based surface fitting framework is applied to reconstruct models from range acquired data. The Delaunay triangulation of the points is chosen as the cell complex and the visibility term is relaxed to accommodate for noise in densely sampled data with scarce line-of-sight information. In addition, a suitable discrete surface quality term is designed and added to the energy. Numerous experiments are presented both on highly deteriorated synthetic data sets and on real-world data sets to assess the robustness of a global optimization based on visibility constraints for surface reconstruction.

Contents

As seen in Chapters and 2, surface reconstruction from range data is driven by numerous applications and has a long research history. While recent methods have exclusively considered unoriented point sets or, on the opposite, prerequired good normal estimation for these points, we only assume the availability of approximate lines of sight: despite being either available or easily recoverable, such datum is often simply thrown away. This additional information is put to use to formulate the reconstruction problem as an energy minimisation on a Delaunay triangulation. Our energy basically measures how well the inside/outside of a Delaunay tetrahedron agrees with soft visibility constraints derived from lines of sight and the likeliness of this labeling as an output surface. Fortunately, our energy can be interpreted as an s-t cut in a special graph allowing a globally optimal labeling of tetrahedra with respect to these constraints and the surface quality measure to be efficiently found as a minimum s-t cut. This simple combination of a Delaunay tetrahedron labeling with the global optimization of a visibility-based energy exhibits strong resilience to various kinds of alterations of the input data.

3.1 Surface reconstruction algorithm

The first step of our method computes a triangulation of the 3D point cloud composed of all merged range images. Each finite vertex of this triangulation comes from one range image, and the relative location of the laser and/or the sensor(s) is (at least, approximately) known. As a consequence, the corresponding line(s) of sight emanating from a vertex to the laser and/or the sensor(s) is required to not cross the reconstructed surface.

3.1.1 Delaunay triangulation

To apply the surface fitting framework of Chapter 2, the chosen cell complex is, as announced, a triangulation of the input points, more specifically its Delaunay triangulation. This choice is motivated by the following remarkable property: under some assumptions, and especially if $\mathcal{P}$ is a "sufficiently dense" sample of a surface, in some sense defined in [Amenta and Bern, 1999], then a good approximation of the surface is "contained" in $\mathrm{Del}(\mathcal{P})$, in the sense that the surface can be accurately reconstructed by selecting an adequate subset of the triangular facets of the Delaunay triangulation which was illustrated at the end of Chapter 2.

While the algorithmic complexity of the Delaunay triangulation of n 3D points is $\mathcal{O}(n^2)$ in general, as recently proven by [Attali et al., 2003], this complexity drops to $\mathcal{O}(n \log n)$ when the points are distributed on a smooth surface (and under a mild uniform sampling condition), which is the case of interest here.

3.1.2 Optimal labeling of tetrahedra

We consider the surface reconstruction problem as a Delaunay tetrahedra labeling problem: tetrahedra are each assigned an inside or outside label. The reconstructed surface, denoted by $\mathcal{S}$ in the following, is therefore a union of oriented Delaunay triangles: it is guaranteed to be watertight and intersection-free as it bounds a volume. We define an energy $\mathrm{E}(\mathcal{S})$, attached to a reconstructed surface $\mathcal{S}$, and gathering two distinct terms:

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{soft-vis}}(\mathcal{S}, \mathcal{P}, v) + \lambda_{\mathrm{qual}}\, \mathrm{E}_{\mathrm{qual}}(\mathcal{S})$$

The term $\mathrm{E}_{\mathrm{soft-vis}}(\mathcal{S}, \mathcal{P}, v)$ is derived from the framework described in Chapter 2 and is a sum of penalties for misalignments and wrong orientations of the surface $\mathcal{S}$ with respect to the constraints imposed by all the lines of sight from the sample points. The term $\mathrm{E}_{\mathrm{qual}}(\mathcal{S})$ penalizes the triangles of $\mathcal{S}$ unlikely to appear on the true surface. λ_{qual} is a positive constant weighting $\mathrm{E}_{\mathrm{qual}}(\mathcal{S})$.

In the next two sections, we present these two energy terms which can be interpreted as costs of s-t cuts on a special graph, allowing our energy to be globally and efficiently minimized with a standard maximum flow algorithm as reminded in Chapter 2. The graph considered is obviously related to the Delaunay triangulation: it has vertices representing the Delaunay tetrahedra and directed edges representing the oriented triangles between adjacent tetrahedra. This graph is augmented with the (abstract) source and sink vertices and with links from each tetrahedron to the source and the sink. The vertices linked to the source correspond to tetrahedra labeled as outside and symmetrically, vertices linked to the sink are inside tetrahedra. The directed edges of a cut are triangles on the oriented surface.

As noted in Chapter 2, the infinite tetrahedra (the tetrahedra lying outside the convex hull of the input points) are also included as vertices in our graph: this allows the labeling to recover open surfaces. Such property is especially useful for outdoor scenes as shown in Section 3.3.

3.1.3 Surface visibility

In this section, the original visibility term of the previous chapter is quickly described in the special case of the Delaunay triangulation of the input points as the underlying cell complex. It is then improved to better cope with scarce visibility information and sample noise.

Let us consider one vertex of the triangulation and one line of sight from this vertex to the laser (or sensor).

Provided the sample position is noise-free, the tetrahedra intersected by this line of sight from this vertex to the sensor or to the laser should be labeled as outside and the tetrahedron behind the vertex should be labeled as inside. By minimizing the number of intersections of this line of sight

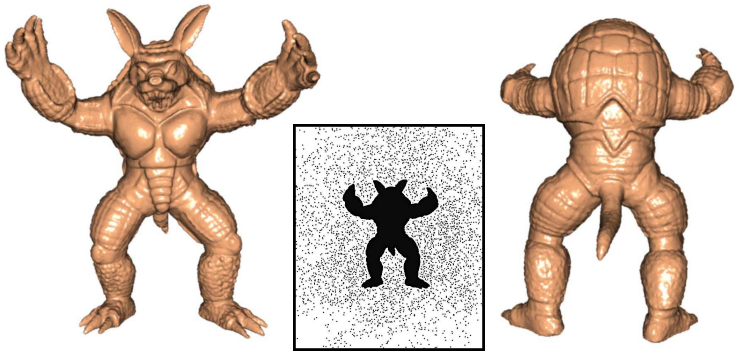


Figure 3.1: ***Stanford armadillo:*** *428K vertices, 841K facets. The original range scans contain many outliers which are automatically eliminated by our method.*

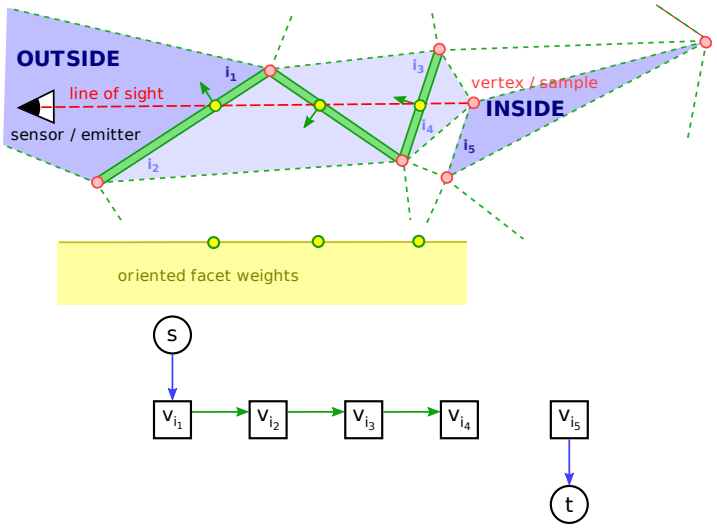
with the oriented surface and penalizing a wrong orientation, we can try imposing this visibility constraint: the triangles crossed by a line of sight from the vertex to the laser (or sensor) are to be penalized. In addition, the surface should also go through the vertex originating the line of sight and the last tetrahedron traversed by the line of sight should be labeled as outside. Let us translate this into weights in the corresponding s-t graph (see Figure 3.2(a)):

1. the left-most darker blue tetrahedron gets an α_{vis}-weighted link to the source (α_{vis} is a positive constant for the line of sight),

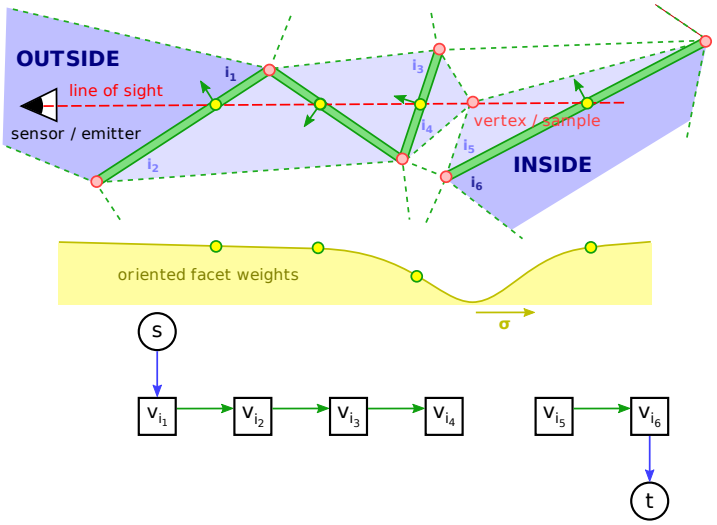
2. the darker green oriented facets on the left of the vertex, crossed by the line of sight and pointing towards the sensor or emitter get an α_{vis}-weighted edge,

3. the darker blue tetrahedron right behind the vertex gets an α_{vis}-weighted link to the sink.

If a confidence measure is available for the line of sight, as indicated in Chapter 2, it should be incorporated into α_{vis}: for instance, [Curless and Levoy, 1996] assigned a confidence value that depends on the angle between the sample normal (evaluated from the range image) and the direction of the line of sight. A surface that goes through the vertex and does not cross the line of sight will not cut any of the weighted edges and links just constructed and will therefore not increase the cost of the s-t cut. This construction is repeated for all available lines of sight of all the vertices of the triangulation by summing their weight contributions: newly generated weights are added to the previously assigned. This can be seen as a kind of "vote" from each line of sight for tetrahedra to be labeled as inside or outside and for oriented triangles to belong to the surface or not. Note that the only tetrahedra that



(a) Visibility



(b) Soft visibility

Figure 3.2: ***Visibility and soft visibility:*** *how a single line of sight (red dotted line) of a vertex of the triangulation (red) from a sample point to a laser (or to a sensor) contributes to the weights assigned to the origin tetrahedron, to the facets it crosses (darker green) and to the final tetrahedra (darker blue).*

get a non zero-weighted link to the source are those (possibly infinite ones) containing the laser sources or sensors optical centers. This integration over hundreds or thousands of thousands of line of sights combined with a global optimization allows our method to exhibit a strong resilience to different kinds of errors in the input data.

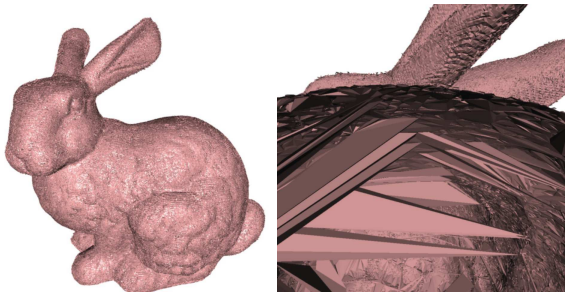


Figure 3.3: ***Stanford bunny (with visibility only):*** *a constant visibility term per line of sight is not suited to the reconstruction of densely (but noisily) sampled surfaces with few lines of sight per sample: it tends to generate bumpy surfaces and mislabels many interior tetrahedra.*

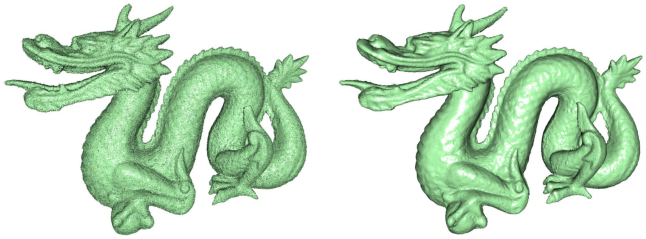


Figure 3.4: ***Stanford dragon (visibility vs. soft visibility):*** *on the left, no tolerance is used and the reconstruction is bumpy and overly complex (1,176K vertices, 2,322K facets), on the right, a reconstruction with tolerance generates a smoother and much coarser mesh (304K vertices, 580K facets).*

While this construction effectively avoids the empty surface solution and the "shrinking bias", it is not completely adapted in the context of reconstruction from range scans which sample a surface very densely (with a noise amplitude close to the sampling density): it tends to generate overly complex surfaces (see Figures 3.3 and 3.4) that are bumpy and have many handles inside the model. The measurement noise found in range image is responsible for the bumpiness of the surface, and the large tetrahedra being mislabeled inside the model appear because each sample point only has one or two line(s) of sight: the tetrahedra that should be labeled as inside because they lie behind a vertex are at a much greater risk of being mislabeled because

Figure 3.5: ***Stanford happy buddha:*** *380K vertices, 738K facets. Setting the tolerance parameter σ too high might create unwanted holes (inside the square).*

no ray from their vertices will ever intersect them. This is precisely what happens in Figure 3.3: some inside tetrahedra of the bunny get mislabeled as outside. In multi-view stereo, this problem can be partly circumvented first by aggregating nearby pair-wise reconstructed 3D points together (merging their line of sight information) as done in the two passive stereo methods of Chapter 1 or by relying on a photo-consistency term to penalize facets. Here, these two problems are solved differently and more elegantly by relaxing the visibility constraints: a tolerance parameter σ is introduced and we modify the edges and links weight constructions. As shown in Figure 3.2(b), the previous construction is extended so that the final tetrahedron on the line of sight does not lie strictly behind the considered vertex but a bit further: it is actually shifted to a distance of 3σ along the line of sight. We also make the oriented facets weights decay with the distance of the intersection of the line of sight with the vertex: each oriented facet intersected by the line of sight of a vertex at a distance d of this vertex gets a weight of $\alpha_{\text{vis}} \left(1 - e^{-d^2/2\sigma^2}\right)$ from this line of sight. As shown in Figure 3.5, the value of σ should be set conservatively. However, changing it reasonably allows to generate more or less complex output meshes (see Figure 3.6). Finally, note that $\sigma = 0$ is equivalent to the first visibility weight construction.

3.1.4 Surface quality

The soft visibility constraints of the previous visibility term are sometimes insufficient to get a good reconstruction: handles might still appear due to tiny elongated tetrahedra being wrongly labeled as outside because few (or no) line of sight intersected them.

At first, a simple heuristic can be used to filter out these tetrahedra

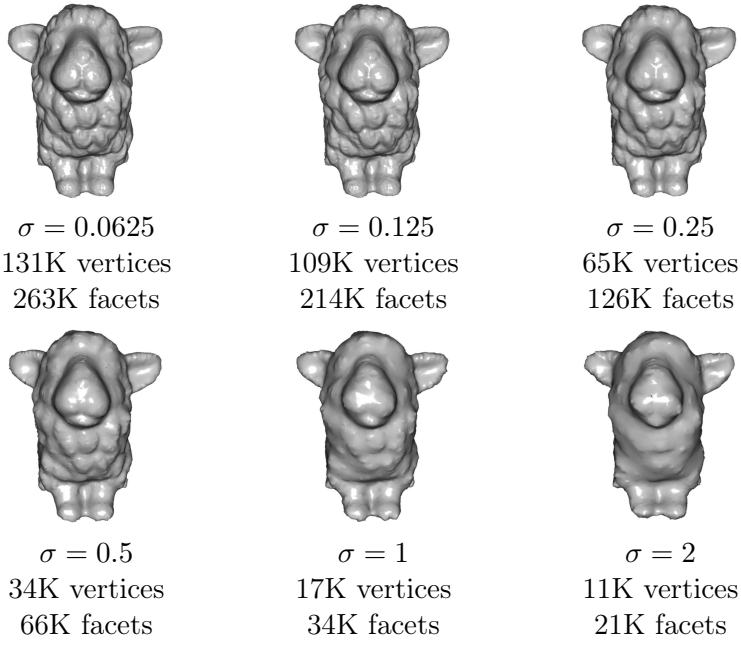


Figure 3.6: ***Influence of the tolerance parameter*** σ *on the reconstruction details and complexity of the* UU sheep.

and ensure the "quality" of the triangles in the output surface: the quality of surface triangle is evaluated as the ratio of the length of their longest edge over the length of their shortest edge (minus one). In our graph, each oriented triangle is weighted with this value. This new quality term tries preventing "badly-shaped" triangles from appearing on the surface.

In practice, this quality measure gives satisfying results and also happens to be used in the second labeling step of [Kolluri et al., 2004]; it is however slightly too discriminating towards skinny triangles that may be required on the surface itself especially in areas where holes in the range images are to be patched (thanks to the Delaunay triangulation).

Instead, we propose to apply a "soft" generalization to 3D of the β-skeleton described in [Amenta et al., 1998a] for curve reconstruction. In 2D, the β-skeleton algorithm computes the Delaunay triangulation of the sample points and chooses the edges of the triangulation whose adjacent triangles have circumcircles centered on opposite sides of the edge and whose radius are both greater than $\beta/2$ times the length of the edge. For dense enough samples, this selection of edges with large empty circumcircles is guaranteed to output a correct reconstruction. Unfortunately, in 3D, almost flat tetrahedra can lie on the surface despite having small empty circumspheres, so the β-skeleton does not generalize well to 3D and may introduce holes. Rather than crudely relabel some tetrahedra selected with a threshold (or even in a

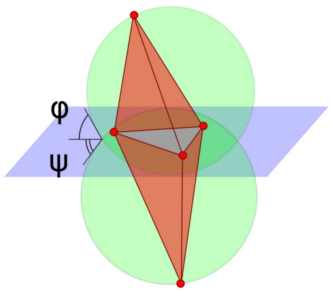


Figure 3.7: ***Soft 3D β-skeleton:*** *a facet of the triangulation, its two adjacent tetrahedra (red) and their circumscribing spheres (green). Their angles ϕ and ψ with the plane (blue) of the facet influence the weight of this facet.*

greedy way), we integrate this quality criterion into our global optimization framework. For a given facet of the triangulation, we consider its two adjacent tetrahedra as pictured in Figure 3.7: the circumscribing spheres of these tetrahedra intersect the plane of the facet at an angle ϕ and ψ. To favor facets with large empty circumspheres, a weight $1 - \min\{\cos(\phi), \cos(\psi)\}$ is added to the two oriented weights of each facet. This way, facets with large empty circumspheres get small penalties for being cut as they are more likely to belong to the surface and conversely, facets with smaller empty spheres are more penalized.

Applying the two constructions described above for the surface visibility term and for the surface quality term assigns oriented edge and link weights to nodes in a directed s-t graph. By computing a minimum s-t cut on this graph, an optimal labeling of tetrahedra with respect to these two combined criteria is obtained and a resulting watertight surface can then be extracted.

For very noisy range scans, any interpolating method may output bumpy surfaces when applied directly to the point cloud. As seen in Section 3.3.2, our method can still be used, at least to help bootstrapping local PDE-based refinements [Whitaker, 1998, Zhao et al., 2001] whose initialization is often problematic. For rendering purpose, in Figures 3.1, 3.8, 3.6, 3.9 and 3.4, at most two steps of the fairing operator of [Vollmer et al., 1999], an enhanced Laplacian smoothing, were applied.

3.2 Implementation details and issues

The presented algorithm was implemented in C++ and relies on the CGAL library [Boissonnat et al., 2000] for the computation of the Delaunay triangulation. The line walk query is described in Appendix B. It also uses Kolmogorov's max-flow algorithm [Boykov and Kolmogorov, 2004] and im-

plementation for the partitioning.

Our current prototype still allows for improvement in both running time and memory use. The max-flow library was designed for efficiency on grid graphs and energies typically used in computer vision. While our network graphs also have fixed connectivity (each node has 4 neighbors), the visibility term in our energy design does not lead to short paths from the source to the sink. Switching to a more adapted max-flow algorithm may significantly improve running times.

Moreover, due to limitations of the max-flow library, edge weights are required to be computed, and only after the whole network graph can be constructed at once. This means that storage for the weights is duplicated. Actually, the graph itself, which can be trivially derived from the Delaunay triangulation, is also stored twice in memory.

Finally, an important increase in memory use can be observed in Table 3.1 between the weights and the minimum s-t cut computations. The algorithm of Kolmogorov's library caches entire search trees which again impacts seriously on the memory footprint.

3.3 Experimental results

Figure 3.8: ***Berkeley angel:*** *361K vertices, 716K facets, genus 3 (the original model has genus 1).*

We tested our method on several (variously sized) publicly available sets of range scans from either the Stanford 3D Scanning Repository (bunny, dragon, armadillo and buddha), the AIM@SHAPE Shape Repository (sheep and elephant) or the U.C. Berkeley Computer Animation and Modeling Group (angel) and also on a new outdoor large-scale data set (rue Soufflot). Only for the Stanford and rue Soufflot data sets a reliable estimation of the laser position and/or direction to the sensor(s) was available. This should however not be seen as a strong limitation since for the other data

Model	#Points	#Tetrahedra	Delaunay	Visibility	Quality	Minimum s-t cut	Overall	
Sheep (Figure 3.6)	153K	966K	2s	3s	1s	3s	10s	$\ll$ 1m
Bunny (Figure 3.12)	362K	2,252K	10s	11s	2s	7s	31s	$<$ 1m
Dragon (Figure 3.4)	1,770K	11,383K	38s	68s	15s	59s	180s	3m
Angel (Figure 3.8)	2,008K	12,637K	41s	86s	16s	48s	190s	3m 10s
Armadillo (Figure 3.1)	2,247K	14,519K	47s	58s	13s	177s	295s	4m 55s
Buddha (Figure 3.5)	2,644K	17,167K	62s	120s	14s	74s	271s	4m 31s
Elephant (Figure 3.9)	4,413K	27,487K	98s	274s	35s	-	-	
Elephant (Figure 3.9) / 64-bits	4,413K	27,487K	93s	189s	32s	102s	417s	6m 57s
rue Soufflot (Figure 3.14) / 64-bits	6,592K	42,062K	176s	416s	40s	521s	1154s	19m 14s
Sheep (Figure 3.6)	153K	966K	48MB	67MB			133MB	
Bunny (Figure 3.12)	362K	2,252K	109MB	154MB			306MB	
Dragon (Figure 3.4)	1,770K	11,383K	543MB	771MB			1.6GB	
Angel (Figure 3.8)	2,008K	12,637K	605MB	858MB			1.7GB	
Armadillo (Figure 3.1)	2,247K	14,519K	690MB	981MB			2.0GB	
Buddha (Figure 3.5)	2,644K	17,167K	815MB	1.1GB			2.4GB	
Elephant (Figure 3.9)	4,413K	27,487K	1.3GB	1.8GB			-	
Elephant (Figure 3.9) / 64-bits	4,413K	27,487K	2.2GB	2.7GB			6.5GB	
rue Soufflot (Figure 3.14) / 64-bits	6,592K	42,062K	3.6GB	5.2GB			9.9GB	

Table 3.1: ***Detailed running time and peak memory use*** *(rounded) of the different steps of the algorithm for the presented reconstructions on an Intel Xeon 3GHz computer (the* Elephant *and* rue Soufflot *data sets required a 64-bits environment to complete the computation). Note: 1GB = 1024MB.*

Model	Sheep (Figure 3.6)		Bunny (Figure 3.12)		Dragon (Figure 3.4)		Angel (Figure 3.8)		Armadillo (Figure 3.1)		Buddha (Figure 3.5)		Elephant (Figure 3.9)		rue Soufflot (Figure 3.14)	
#Points	153K		362K		1,770K		2,008K		2,247K		2,644K		4,413K		6,592K	
								Running time								
Our algorithm	10s		31s		180s		190s		295s		271s		417s		1154s	
Poisson	19s	[8]	65s	[9]	180s	[10]	363s	[10]	208s	[11]	602s	[11]	566s	[11]	1058s	[13]
	20s	[9]	66s	[10]	461s	[11]	376s	[11]	751s	[12]	546s	[12]	934s	[12]	1608s	[14]
								Peak memory use								
Our algorithm	133MB		306MB		1.6GB		1.7GB		2.0GB		2.4GB		6.5GB		9.9GB	
Poisson	237MB	[8]	300MB	[9]	655MB	[10]	785MB	[10]	728MB	[11]	1.2GB	[11]	1.1GB	[11]	3.3GB	[13]
	239MB	[9]	311MB	[10]	975MB	[11]	856MB	[11]	1.4GB	[12]	1.2GB	[12]	1.7GB	[12]	9.5GB	[14]

Table 3.2: ***Comparison of running time and peak memory use*** *of the surface reconstruction algorithm with Poisson surface reconstruction [Kazhdan et al., 2006] for the presented reconstructions (maximum octree depth in brackets). The reconstructions of the* Elephant *and* rue Soufflot *data sets were run in a 64 bits environment. Note: 1GB = 1024MB.*

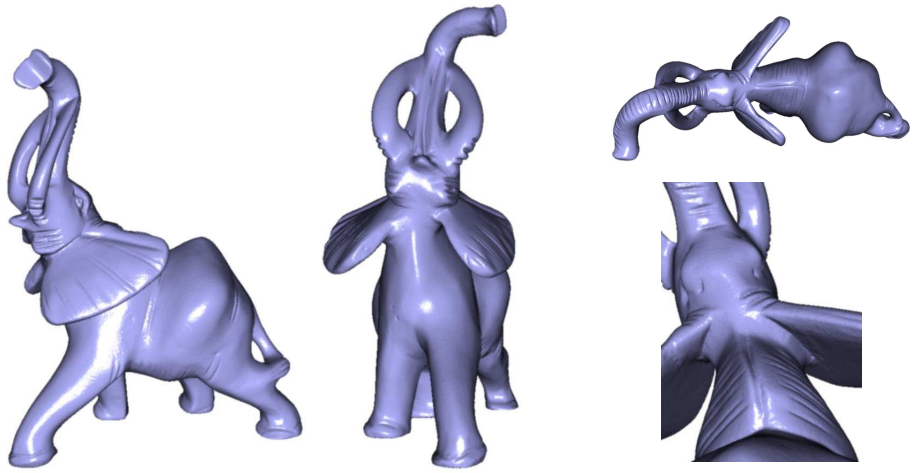


Figure 3.9: ***INRIA/ISTI Livingstone elephant:*** *821K vertices, 1,586K facets.*

sets, we used less precise, approximate lines of sight and this did not result in significant artefacts in the reconstructions (the more strongly penalized facets and tetrahedra lies inside or far outside the object). Moreover, even if lines of sight can not be reliably guessed, the hidden point removal operator of [Katz et al., 2007] could potentially be applied from virtual laser positions to recover such visibility information, at least for properly sampled data without outliers.

In all the experiments, the same value of 5 for λ_{qual} was used to balance visibility and quality. Instead of weighting lines of sight with a confidence estimation as in [Curless and Levoy, 1996] and suggested in Section 3.1.3, α_{vis} was purposefully fixed to a constant 32. These two constants were heuristically found on one data set and kept for all the presented results. Finally, the tolerance σ which is supposed to reflect the expected noise level in the data was uniformly set on a per model basis but estimated the same way for every model (1/2 of the median range grid diagonal).

As indicated in Table 3.1 and despite the shortcomings of our implementation, our method proves to be fast and scales well (almost linearly) with the size of the input point cloud both in running time and peak memory use (much better than spectral methods): this might be a hint that modeling the surface reconstruction problem by taking into account as much information as possible about the scanning process actually leads to a better posed minimization problem. A comparison with Poisson surface reconstruction [Kazhdan et al., 2006], run at two successive octree depths, shows in Table 3.2 that our method is competitive in running time but tends to require more memory due to both its interpolatory nature and the known limits of our implementation.

In addition to the previously shown reconstructions, we highlight in the next three sections the robustness of our approach compared to several other methods: the Delaunay-based local algorithms Robust Cocone [Dey and Goswami, 2006] and Power Crust [Amenta et al., 2001], and the implicit methods Adaptive Compactly Supported Radial Basis Functions [Ohtake et al., 2003] and Poisson surface reconstruction [Kazhdan et al., 2006]. The two implicit methods require oriented normal estimates. While more elaborate methods exist (see [Dey et al., 2005] for a recent study), these normals were computed in each scan by fitting a plane to the neighbors of each sample with least-squares weighted by a standard radial function of the distance [Wendland, 1995]. The correct normal orientation is found using the lines of sight. The normals were obviously estimated from the modified data of the experiments to make the comparison fair. More involved methods should not be expected to provide much more precise normal estimations for the different altered data used in the experiments.

Finally in the last section, Poisson surface reconstruction and our approach are challenged on a difficult large-scale outdoor scene.

3.3.1 Robustness to non-uniform sampling

Figure 3.10 illustrates the adaptivity of our method to a non-uniform sampling of the surface (plus a decent amount of measurement noise): a plane partitions the input point set in halves and one of these halves is heavily downsampled. While Robust Cocone seriously degrades when a 128× undersampling is reached, from the beginning, Power Crust splits the two front paws of the sheep. It better handles the undersampling until a 1024× downsampling when its local approach make some important details disappear on the front part of the sheep: the right ear fades away, and the bottom right part around the paw almost vanishes. Poisson progressively shrinks the right part of the sheep, losing all features and Adaptive CS RBF is quickly in trouble. By relying on the visibility information available from the scarce samples, our method is still able to reconstruct a surface that resembles the original model.

3.3.2 Robustness to noise

While all the data sets used to present our result already contain various amounts of measurement errors, we provide further evidence of the ability of our method to cope with severe amounts of noise. In Figure 3.11, we adopt a protocol analogous to [Kolluri et al., 2004] which add isotropic Gaussian noise to the original point coordinates in the Stanford bunny scans. We instead add anisotropic Gaussian noise along the laser line of sight only. Exceptionally, we present results not only with the usual fixed tolerance parameter σ but also results with a varying value of σ matching the amount of

added noise. The standard deviation of the added noise is measured in terms of the median length $d = 0.001$ of the range grids diagonals. Point clouds altered with a noise of deviation several times this length are extremely fuzzy, and a correct reconstruction is hardly expected. Holes quickly appear at $+0.8d$ with Power Crust. Robust Cocone which is designed to cope with reasonable amount of noise quickly loses the features of the model (ears, neck and front paws) between $+0.4d$ to $+0.9d$. Adaptive CS RBF is unable to handle the additional noise. Poisson, however, is extremely resistant and still reconstructs a smooth surface at very high levels of noise. It nevertheless begins to seriously degrade after $+2d$. Our method still outputs a genus 0 (albeit bumpy) surface with deviation $+2d$, after this point, the ears of the model begin to fade away and after a $+3d$ deviation the reconstruction irreversibly but slowly degrades. Our method with an adapted σ outputs a much smoother reconstruction and degrades more gracefully.

3.3.3 Robustness to outliers

As illustrated in Figure 3.1, range scans usually contain some outliers. On a synthetic example of 26K noise-free points, the spectral surface reconstruction method of [Kolluri et al., 2004] was shown to handle 1,200 outliers (or 4.5% of outliers) without any degradation, it then slightly degrades with 1,800 outliers (6.5%) and completely disintegrates with about 10,000 outliers (28%). Here, outliers are added to the original data in much larger amount or ratio. In Figures 3.12 and 3.13, we show how the results of other algorithms and ours degrade as randomly generated outliers are gradually added to the 362K points of the Stanford bunny (in fact, we are showing robustness to measurement noise *and* synthetic outliers): the outliers are added scan per scan, their position projects to the range grid and their location is randomly chosen within the bounding box of the range image. This protocol effectively simulates outliers generated during the acquisition. All other tested methods are defeated earlier than ours and unable to recover any useful reconstruction. Poisson reconstruction is the strongest contender, but the estimated oriented normal field tends to be inconsistent at outliers and this may actually help this method to filter out outliers.

It is nevertheless pleasant to observe that taking into account the visibility information from the scanning process allows our method to deal with an impressive number of outliers (up to 850,000 or $\sim$70% of outliers) with only very slight degradation of the recovered surface (a handful of outside tetrahedra might be mislabeled now and then). We have consistently observed that our reconstructions irreversibly degrade only when the number of outliers begins to exceed twice the number of inliers: this definitely confirms the suitability of a global optimization based on lines of sight for outlier removal. While such massive amounts of outliers are not realistic for laboratory acquisitions, our method can be applied with success to outdoor range data

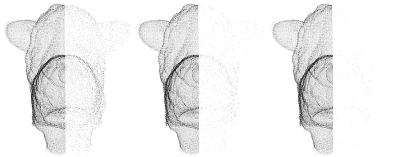
or to quasi-dense point clouds from image matching or video tracking that would contain some amount of outliers.

3.3.4 Large-scale outdoor range data

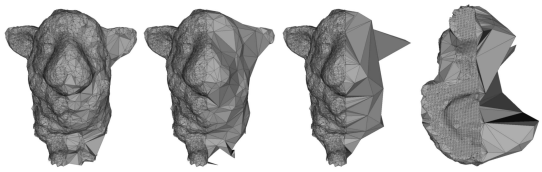
In this section, we show the result of our reconstruction algorithm compared to Poisson surface reconstruction [Kazhdan et al., 2006] on a challenging data set. The range data was acquired in the rue Soufflot in Paris while driving a mobile vehicle equipped with a time-of-flight range finder paired with a GPS/IMU unit which automatically registers the acquired data (the vehicle was also equipped with several cameras). The images and range data are provided courtesy of MATIS[1], French Mapping Agency (IGN). This datum is particularly difficult: it hardly meets satisfying sampling conditions, it includes moving objects (the pedestrians and the other vehicles), features are present at a wide range of different scales (the street is about 250 meters long) and many parts are occluded. In addition the whole point cloud weights 6.7M samples.

While the presented results would probably require some post-processing before any use in applications, they still demonstrate the potential of our approach even on data acquired without controlled scanning conditions. The running time and peak memory use of our method are about 16m and 9.9GB. Poisson surface reconstruction was executed at the largest possible depth (14) on the same machine and takes 26m and 9.5GB of memory. Recently, extension of previous reconstruction algorithms have been proposed to handle massive data sets with limited memory [Fiorin et al., 2007, Allègre et al., 2007, Bolitho et al., 2007]. Since such considerations are out of scope in the present chapter, only the original (non out-of-core) algorithm of Poisson surface reconstruction was used. As shown in Figures 3.14, 3.15, 3.16 and 3.17, our method reconstructs the whole open scene with very thin details (however, for illustration purposes, large triangles close to the convex hull had to be filtered out from the reconstruction by thresholding). Poisson reconstructs a closed scene (which thus required editing) that is less complete than ours (the adjacent streets are much less extended and the roofs are missing, see Figures 3.14 and 3.15). It also tends to smooth out the fine structures our method is able to recover (Figures 3.16 and 3.17).

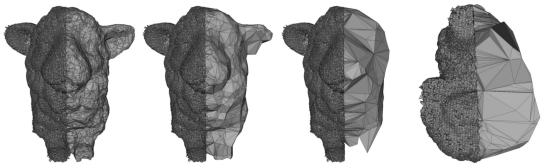
[1] http://recherche.ign.fr/labos/matis/.



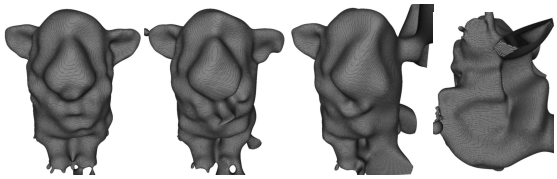
(a) Point cloud (right half downsampled by a factor 16, 128 and 1024)



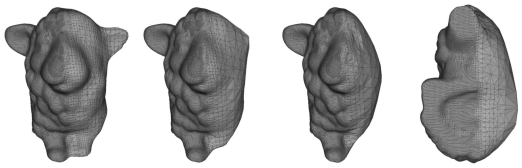
(b) Robust Cocone



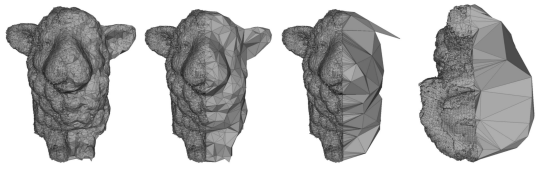
(c) Power Crust



(d) Adaptive CS RBF



(e) Poisson



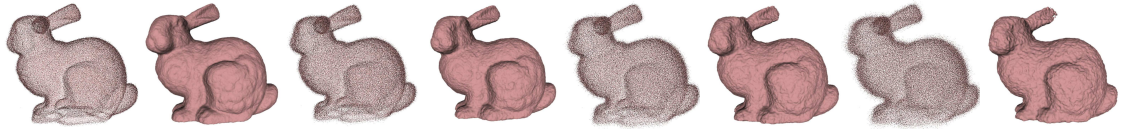
(f) Our method

Figure 3.10: ***Robustness to undersampling:*** *the right-most bottom view corresponds to a 1024× downsampling.*

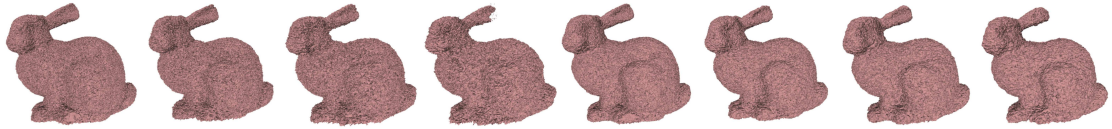
(a) Power Crust: $+0.1d$ and $+0.8d$

(b) Robust Cocone: $+0.4d$ and $+0.9d$

(c) Adaptive CS RBF: $+0.3d$ and $+0.5d$



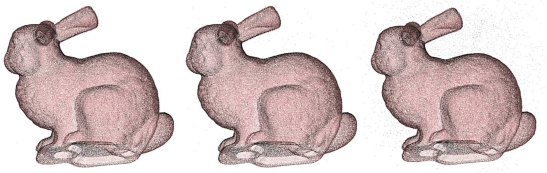
(d) Poisson: $+0.9d$, $+1.5d$, $+2d$ and $+3d$



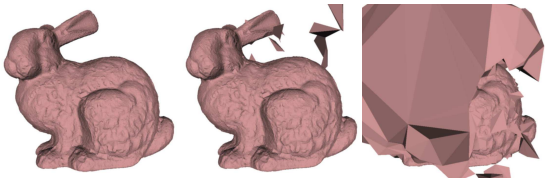
(e) Our method with fixed σ: $+0.9d$, $+1.5d$, $+2d$ and $+3d$

(f) Our method with varying σ: $+0.9d$, $+1.5d$, $+2d$ and $+3d$

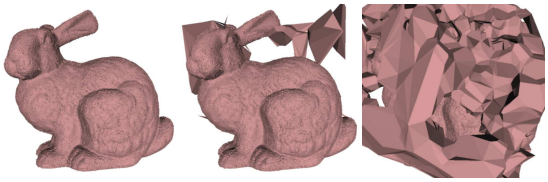
Figure 3.11: ***Robustness to noise:*** *the input point cloud is on the left of the first corresponding reconstruction result.*



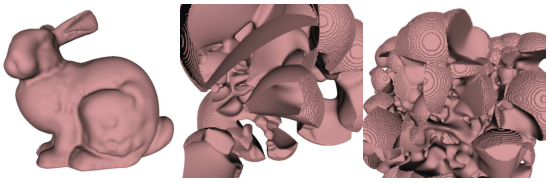
(a) Point cloud plus 0, 200 and 4,000 outlier(s)



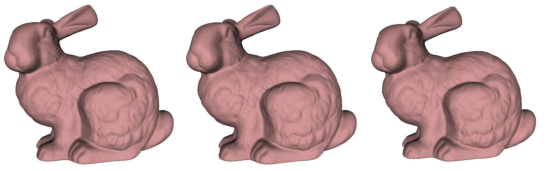
(b) Robust Cocone



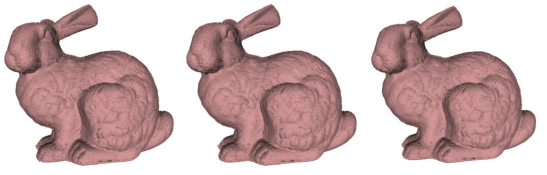
(c) Power Crust



(d) Adaptive CS RBF

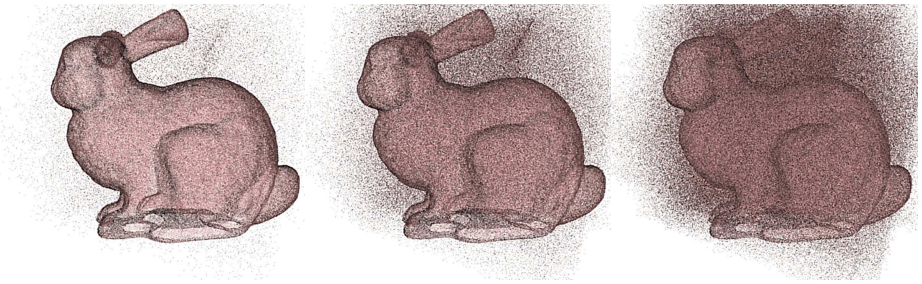


(e) Poisson



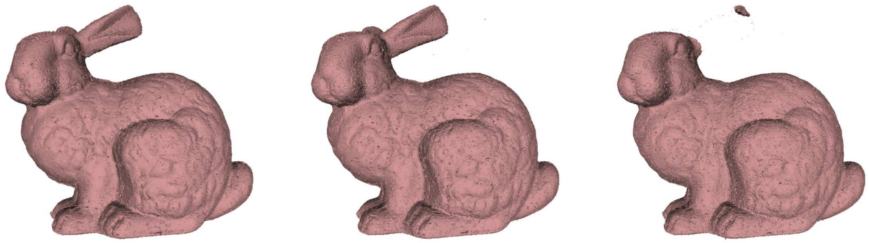
(f) Our method

Figure 3.12: ***Robustness to relatively few outliers.***



(a) Point cloud plus 50,000, 300,000 and 850,000 outliers

(b) Poisson



(c) Our method

Figure 3.13: ***Robustness to large amounts of outliers.***

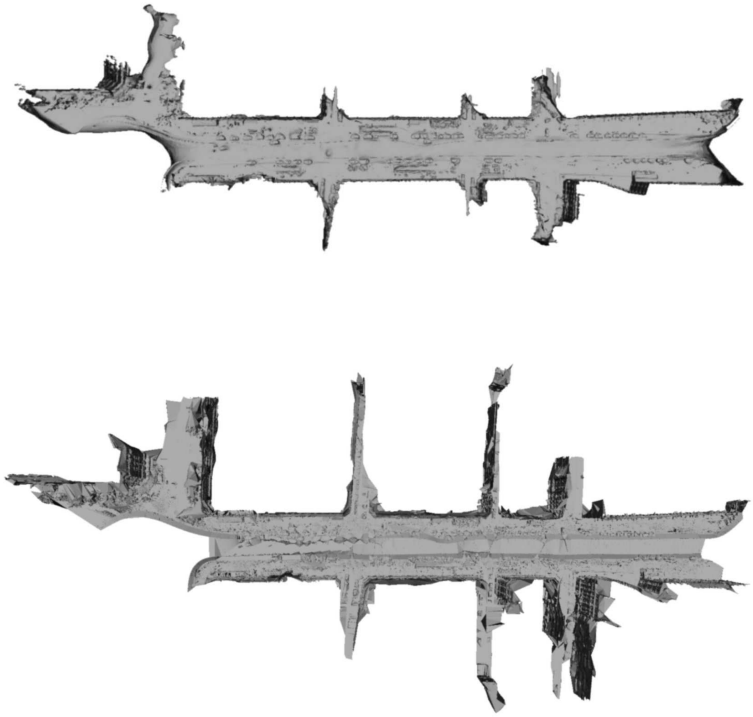


Figure 3.14: ***Top view of the*** **rue Soufflot*:*** *reconstruction results for Poisson (top) and our method (bottom).*

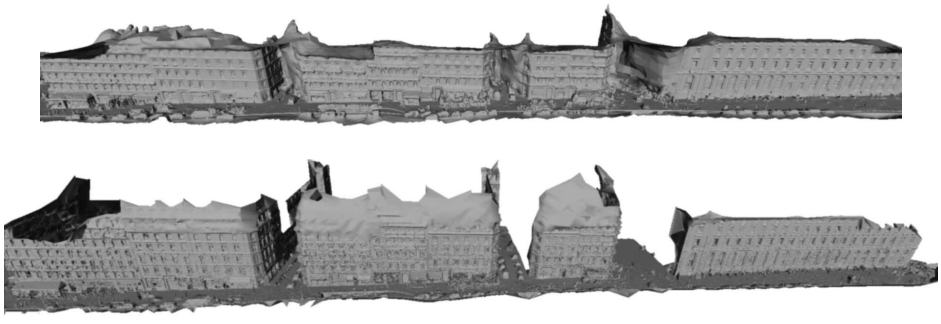


Figure 3.15: ***Panorama view of the*** **rue Soufflot*:*** *reconstruction results of Poisson (top) and our method (bottom).*

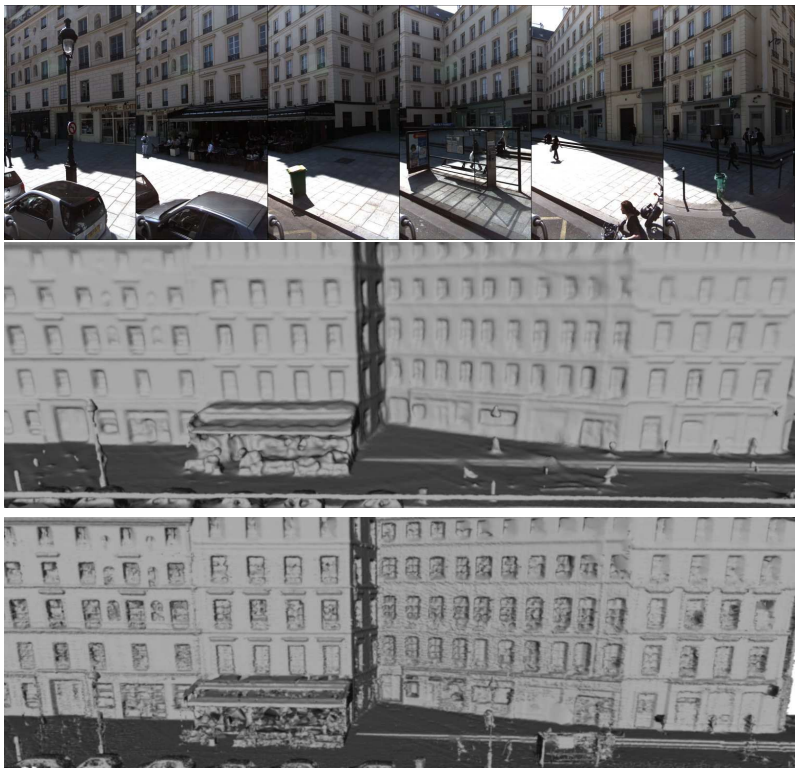


Figure 3.16: ***Reconstruction details of the* rue Soufflot*:*** *acquired images (top), corresponding reconstruction results of Poisson (middle) and our method (bottom).*

Figure 3.17: ***Reconstruction details of the* rue Soufflot*:*** *acquired images (left), corresponding reconstruction results of Poisson (center) and our method (right).*

CHAPTER 4

Multi-view stereo reconstruction of large-scale scenes

While the precision of image-based modeling methods has drastically increased in the past few years, most dense multi-view stereovision methods are still not applicable to large-scale outdoor scenes. A variant of the visibility-based optimization on Delaunay tetrahedra is used in a multi-view stereo pipeline as the key initialization of a mesh-based photometric variational refinement. Qualitative and quantitative evaluation of the whole pipeline is done on a number of scenes showing its results to compare very favorably with the current state-of-the-art.

Contents

Boosted by the Middlebury challenge [Seitz et al., 2006], the precision of dense multi-view stereovision methods has increased drastically in the past few years. Yet, most methods, although performing well on this benchmark, are still inapplicable to large-scale data sets taken under uncontrolled conditions. This chapter proposes a multi-view stereo pipeline which deals with very large scenes while producing highly detailed reconstructions within very reasonable time. The keys to these benefits are twofold: 1. a minimum *s-t* cut based global optimization that transforms a dense point cloud into a visibility consistent mesh, followed by 2. a mesh-based variational refinement that captures small details, smartly handling photo-consistency, regularization and adaptive resolution. The method has been tested on numerous large-scale outdoor scenes. The accuracy of its reconstructions is also measured on the recent dense multi-view benchmark proposed in [Strecha et al., 2008], showing the results to compare more than favorably with the current state-of-the-art.

4.1 Introduction

Motivation As noted in the introduction chapter, scene reconstruction from multiple images has always been an active field of research in computer vision. This problem finds many practical applications in the entertainment industry, in earth sciences and in cultural heritage digital archival. When highly detailed reconstructions are needed, the combination of laser-based acquisition and standard surface reconstruction are usually applied successfully. However, as reminded in Chapter 1, these methods and the acquisition process are rather complex to set for large-scale outdoor reconstructions, particularly when aerial acquisition is required (see for instance the recent detailed reconstruction of the Bayon temple in Angkor [Banno et al., 2008] with range finders attached to flying balloons). A long-term goal is to replace these methods with image-based ones, yielding considerable savings both in time and money. Recent advances in multi-view stereo methods made this goal closer than ever.

Multi-view stereo Since the review of [Seitz et al., 2006] and the associated Middlebury evaluation, a lot of research has been focusing on multi-view reconstruction of small objects with tightly controlled imaging conditions. This has led to the development of many algorithms whose results are beginning to challenge the precision of laser-based reconstructions. However most of these algorithms are not directly suited to large-scale scenes. A number of multi-view stereo algorithms have been proposed that exploit the *visual hull* of [Laurentini, 1994]. Assuming the foreground object can be separated from the background, 2D silhouettes delimiting the object of interest are extracted in each images. Their back-projections in 3D yield a set of gener-

Figure 4.1: ***Visual hull.*** *Objects acquired under controlled imaging conditions from Seitz et al. [2006], silhouetted images and the corresponding (approximated) visual hulls. While the visual hull does not capture deep concavities (the temple's stairways or the dino's legs, for instance), it already provides a good initialization for variational methods.*

alized cones that intersects in the *visual-hull* (see Figure 4.1). Many dense multi-view methods rely on this information either as an initial guess for further optimization [Hernández and Schmitt, 2004, Furukawa and Ponce, 2006, Hornung and Kobbelt, 2006a,b, Starck et al., 2006, Tran and Davis, 2006, Vogiatzis et al., 2007, Yu et al., 2006], as a soft constraint [Hernández and Schmitt, 2004, Kolev et al., 2009] or even as a hard constraint [Sinha and Pollefeys, 2005, Furukawa and Ponce, 2006] to be fulfilled by the reconstructed shape.

While the unavailability of the visual hull discards many of the top-performing multi-view stereo algorithms of the Middlebury challenge of Seitz et al. [2006] for our purpose, the requirement for the ability to handle large-scale scenes discards most of the others, in particular volumetric methods, *i.e.* methods based on a regular decomposition of the domain into elementary cells, typically voxels. Obviously, this approach is mainly suited to compact objects admitting a tight enclosing box, as its computational and memory costs quickly become prohibitive when the size of the domain increases. This includes space carving [Seitz and Dyer, 1999, Kutulakos and Seitz, 2000, Broadhurst et al., 2001, Yang et al., 2003, Treuille et al., 2004], level sets [Faugeras and Keriven, 1998, Jin et al., 2005, Pons et al., 2007a], and volumetric graph cuts [Vogiatzis et al., 2005, Boykov and Lempitsky,

2006, Hornung and Kobbelt, 2006b, Lempitsky et al., 2006, Starck et al., 2006, Tran and Davis, 2006] (though [Sinha et al., 2007, Hernández et al., 2007] propose regular volumetric grid adaptive to photo-consistency measures to push the resolution limit further).

Finally, cluttered scenes disqualify variational methods [Faugeras and Keriven, 1998, Hernández and Schmitt, 2004, Duan et al., 2004, Jin et al., 2005, Lhuillier and Quan, 2005, Pons et al., 2007a, Delaunoy et al., 2008] that can easily get stuck into local minima, unless a way to estimating a close and reliable initial guess is provided that takes visibility into account.

Large-scale multi-view stereo Multi-view stereo methods that have proved to be more adapted to large-scale scenes (*e.g.* outdoor architectural scenes) are those representing the initial geometry with sparser measurements as depth maps or point clouds.

The performance of depth maps based methods [Kolmogorov and Zabih, 2002, Strecha et al., 2003, 2004, Gargallo and Sturm, 2005, Strecha et al., 2006, Goesele et al., 2006, 2007, Tylecek and Sara, 2009] for complete reconstruction however seems to be lower than previously discussed approaches, either as regards accuracy or completeness of the obtained model. This may be due to the merging process and to the difficulty to take visibility into account globally and consistently. While visibility is taken into account to fuse depth maps in [Merrell et al., 2007], the focus on high performance prevents the use of a global optimization. Recently, [Zach et al., 2007] proposed a globally optimal variational merging of truncated signed distance maps using a volumetric grid. Another exception could be the work of [Campbell et al., 2008], currently one of the most accurate method according to the Middlebury evaluation, but this method relies on a volumetric graph cut [Hernández et al., 2007] that cannot handle large-scale scenes.

Large-scale and high-resolution multi-view stereo In contrast to these depth maps based methods, the authors of [Furukawa and Ponce, 2007] propose a very accurate reconstruction that generates and propagates a semi-dense set of patches. This method has shown impressive results but relies on heuristics to filter and expand sets of oriented patches. The final surface reconstruction is done with Poisson surface reconstruction [Kazhdan et al., 2006] which requires dense and uniformly sampled point clouds and does not handle visibility issues.

This method was tested on the large-scale data sets provided by Christoph Strecha *et al.* [Strecha et al., 2008], the only available evaluation that allows comparison on large-scale scenes (to our knowledge). The results of [Furukawa and Ponce, 2007] were so far significantly more accurate and complete than the few other submitted ones. Note that the relatively small number of competitors *w.r.t.* the Middlebury challenge corroborates the above discus-

sion about the non-ability of most of the methods to handle such data. This quantitative challenge will be relied upon to demonstrate the superiority of our method. Results on other data sets confirm this superiority by visual qualitative evaluation.

Our multi-view stereo method consists in a pipeline that handles large-scale scenes while providing very accurate reconstructions. The whole pipeline is designed to not sacrifice accuracy for scalability. Several design choices are made and justified by an analysis of the weaknesses of previous methods. The pipeline consists of three main steps:

1. the generation of a quasi-dense point cloud with the standard passive multi-view stereo techniques described in Chapter 1,
2. the extraction of a mesh that respects visibility constraints and is close to the final reconstruction, with a minimum s-t cut-based optimization using the surface fitting framework of Chapter 2 over the Delaunay triangulation of the points (similarly to the previous chapter),
3. the variational refinement of this initial mesh to optimize its photo-consistency.

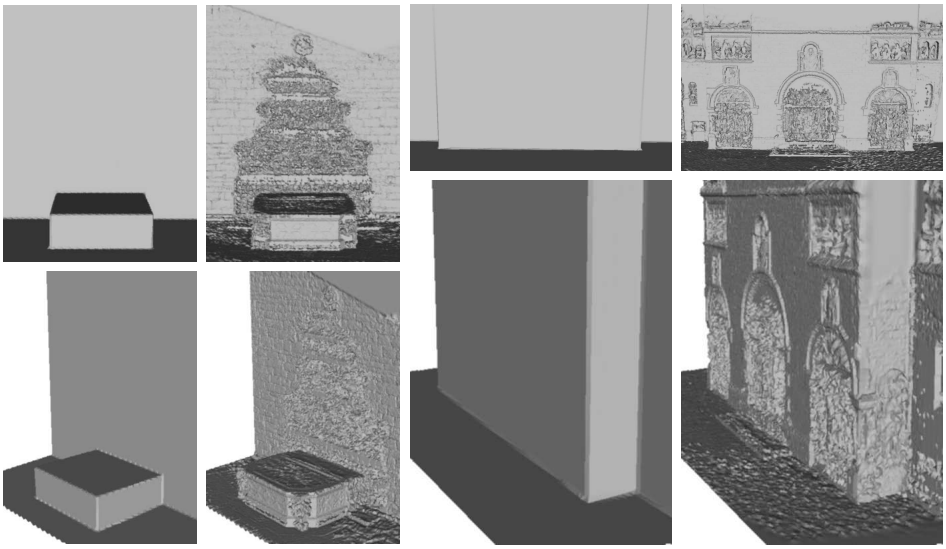


Figure 4.2: ***Bad initializations.*** *The photometric variational refinement requires a good initialization: here, a very crude initial surface was used on the* fountain-P11 *and* Herz-Jesu-25 *data sets of [Strecha et al., 2008].*

4.2 Quasi-dense point cloud

As described in Chapter 1, a quasi-dense point is generated either from interest points or from plane sweeping. Close points are then aggregated

efficiently using the Delaunay triangulation or clustering, so that a point of the final cloud originates from possibly more than two images. The overall process is fast, the NCC (and, if necessary, the plane sweep) being easily implemented on graphics hardware.

Above all, relying on thresholds and possibly generating numerous outliers is not a serious concern. The only goal of this step is to generate enough points so that the following global optimization finds a close enough surface from the tetrahedron facets.

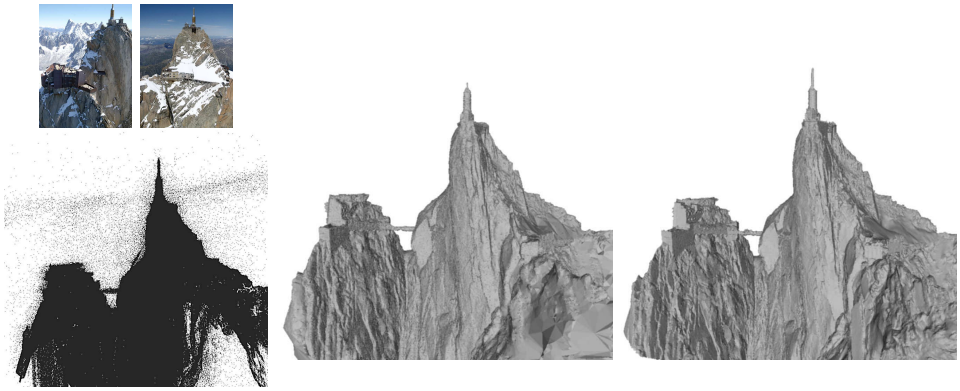


Figure 4.3: ***Aiguille du Midi data set.*** *From left to right: two of the* 53 *images (*10 *Mpix) of a mountain summit taken from a helicopter (© B.Vallet/IMAGINE), point cloud from interest points, initial surface* $\mathcal{S}^0$ *(smoothed for rendering purpose), final reconstruction (*600,000 *triangles).*

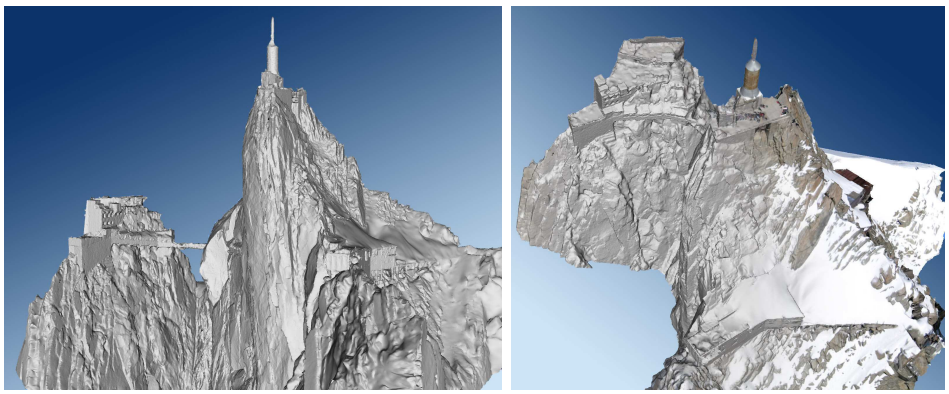


Figure 4.4: ***Aiguille du Midi data set.*** *Rendered views of the reconstruction of the Aiguille du Midi data set without and with texture remapping using [Allène et al., 2008].*

4.3 Initial mesh

Early work in stereovision [Faugeras et al., 1990] already relied on constrained Delaunay triangulations of robustly matched segments to reconstruct environments. Here, the second step of our multi-view pipeline consists in producing from the Delaunay triangulation of the generated quasi-dense point cloud a mesh taking visibility into account which is accurate enough to be then refined by a variational optimization. The hypothesis is then that the gradient descent of some energy deforms this mesh into a local minimum considered as the final reconstruction. The correct initialization of this local variational optimization is thus critical as shown in Figure 4.2.

Our visibility-based surface fitting framework of Chapter 2 perfectly fulfills this purpose for cluttered large-scale scenes. From the image-based point cloud where each point memorizes the two or more images from which it has been triangulated (as described in Chapter 1), the Delaunay triangulation of these points is built. Then, the Delaunay tetrahedra are labeled inside or outside the object so that the labeling minimizes some energy and finally the surface is extracted as the set of triangles between inside and outside tetrahedra. The energy takes visibility into account and exactly follows the $\mathrm{E}_{\mathrm{vis}}$ term of Chapter 1: each ray from a vertex of the Delaunay triangulation to the cameras from which it has been generated is enforced to intersect the oriented output surface as few times as possible. Note that since the noise model for stereo is more complex than for range data and that several lines of sight per point are available, the relaxed visibility term $\mathrm{E}_{\mathrm{soft-vis}}$ of the previous chapter is not used as it requires an additional tolerance parameter σ that could not be set as appropriately as in the previous chapter.

However, as input images are available, an additional photo-consistency term $\mathrm{E}_{\mathrm{photo}}$ may be used to favor surfaces with best matching re-projections in the different views. This can also be implemented within the minimum s-t cut framework of Chapter 2: the edge of each oriented facet F gets a weight $1 - \rho(F)$ where $\rho(F) \in [0, 1]$ is a the estimated photo-consistency cost of the oriented facet F. The photo-consistency cost $\rho(F)$ can be computed as follows. F being an oriented facet of the triangulation, its vertices have associated visibility information that can be used to determine the most fronto-parallel reference view i. Pairwise photo-consistencies $\rho_{ij}(F)$ of the oriented facet in this reference view i and another view j are then calculated and their median is finally chosen as $\rho(F)$: $\rho(F) = \mathrm{med}\{\rho_{ij}(F)\}_j$ with $\rho_{ij}(F)$ defined for instance as the mean normalized cross-correlation on a smooth Gaussian window sliding over the projection of the facet F in the reference view (as in Chapter 1).

The resulting point cloud is very dense compared to standard passive stereo and typically contains millions of points (see Figure 4.3). The visibility term of the energy of Chapter 2 is very effective to filter out outliers from stereo point clouds. Due to the high density of point clouds from reg-

ular grids or interest points, triangles lying near the surface are very small and we experimented with such complimentary photo-consistency term: the output surface is less noisy and this term may help removing structured outliers. However, since the output surface is only used as an initialization for a variational photometric refinement, this term is advantageously replaced with the simple surface quality term $\mathrm{E}_{\mathrm{qual}}$ of the previous chapter used for surface reconstruction from range scans. This term penalizes facets unlikely to appear on a densely sampled surface. As a result and contrarily to the original method of [Labatut et al., 2007], the minimum s-t cut step encodes discrete visibility and surface quality, saving an appreciable amount of time (the photo-consistency has to be computed on *all* the facets of the Delaunay triangulation). Support for infinite tetrahedra is also added (tetrahedra with one facet on the convex hull and incident to the infinite vertex). This not only allows the observer to be "inside" the object, but also makes it possible to generate open meshes. This is an important aspect for outdoor scenes.

The energy to label tetrahedra which can be globally minimized with minimum s-t cut, is thus:

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}}(\mathcal{S}, \mathcal{P}, v) + \lambda_{\mathrm{qual}}\, \mathrm{E}_{\mathrm{qual}}(\mathcal{S}) \ ,$$

where $\mathcal{P}$ is the generated point cloud and v the associated visibility sets. The resulting initial mesh is denoted by $\mathcal{S}^0$.

4.4 Variational refinement

The obtained mesh $\mathcal{S}^0$ is still noisy and does not capture fine details. It is refined with all the image data, using a variational multi-view stereovision approach pioneered by [Faugeras and Keriven, 1998]: $\mathcal{S}^0$ serves as the initial condition of a gradient descent of an adequate energy function. As $\mathcal{S}^0$ is close to the desired solution, this local optimization is very unlikely to get trapped in an irrelevant local minimum. The energy function and optimization procedure is now justified, by presenting the numerous improvements to the initial method. These are not details, or more accurately, to get a detailed reconstruction, every detail matters.

4.4.1 Photo-consistency

Let $\mathcal{S}$ be the object surface, $\boldsymbol{x}$ a point on $\mathcal{S}$, $\overrightarrow{n}$ the normal to $\mathcal{S}$ at point $\boldsymbol{x}$, $g_{ij}(\boldsymbol{x}, \overrightarrow{n})$ a positive decreasing function of a photo-consistency measure of the patch $(\boldsymbol{x}, \overrightarrow{n})$ according to images i and j, and $v_{ij}^{\mathcal{S}}(\boldsymbol{x}) \in \{0, 1\}$ the visibility of $\boldsymbol{x}$ in these images according to $\mathcal{S}$. The original energy in [Faugeras and Keriven, 1998] is:

$$\mathrm{E}_{\mathrm{photo}}(\mathcal{S}) = \sum_{i,j} \int_{\mathcal{S}} v_{ij}^{\mathcal{S}}(\boldsymbol{x})\, g_{ij}(\boldsymbol{x}, \overrightarrow{n})\, \mathrm{d}\mathcal{S} \ . \tag{4.1}$$

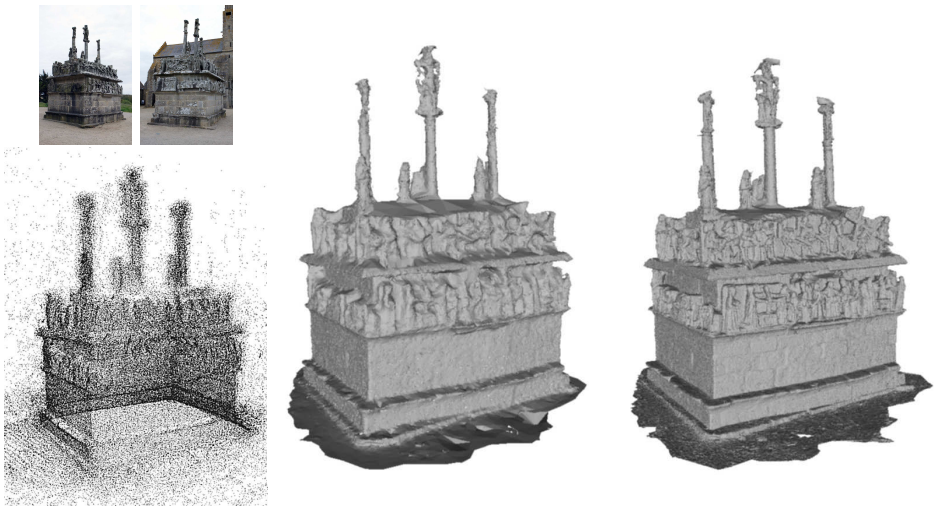


Figure 4.5: ***calvary data set.*** *From left to right: two of the* 27 *images (*8 *Mpix) of a sculpted calvary taken from the ground, the generated point cloud (*$1,300,000$ *points), the initial mesh* $\mathcal{S}^0$ *(*$1,200,000$ *triangles) and the final reconstruction (*$1,850,000$ *triangles).*

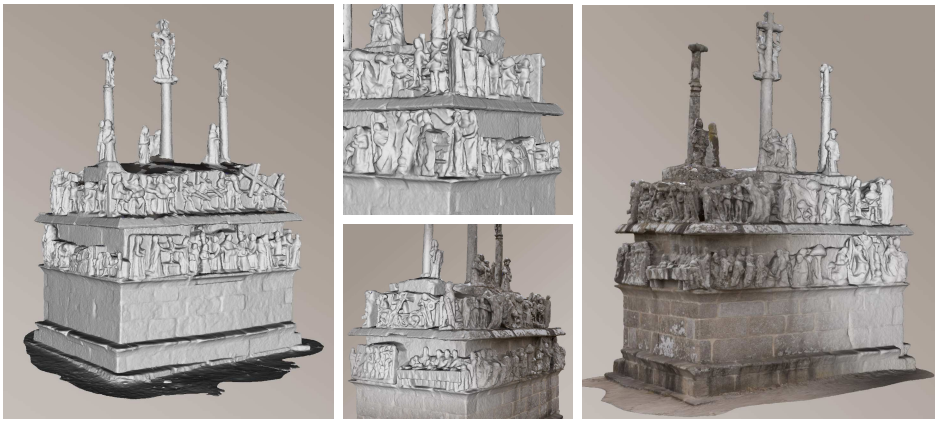


Figure 4.6: ***calvary data set.*** *Rendered and close views of the reconstruction of the calvary data set without and with texture remapping using [Allène et al., 2008].*

To this energy, the re-projection error introduced by [Pons et al., 2007a] is preferred, namely:

$$\mathrm{E}_{\mathrm{error}}(\mathcal{S}) = \sum_{i,j} \int_{\Omega_{ij}^{\mathcal{S}}} h(I_i, I_{ij}^{\mathcal{S}})(\boldsymbol{x}_i)\, \mathrm{d}\mathcal{S}_i \ , \tag{4.2}$$

where $h(I, J)(\boldsymbol{x})$ is a positive decreasing function of a photo-consistency measure between images I and J at pixel $\boldsymbol{x}$, $I_{ij}^{\mathcal{S}} = I_j \circ \Pi_j \circ \Pi_i^{-1}$ is the re-projection

of image I_j into image I_i induced by $\mathcal{S}$ and $\Omega_{ij}^{\mathcal{S}}$ is the domain of definition of this re-projection. This energy measures, for each considered camera pair, the dissimilarity between the portion of a reference image corresponding to the projected surface and a portion of another image re-projected via the surface into the reference image.

This summation has three major advantages over the original one:

1. re-projecting I_j into I_i according to $\mathcal{S}$ uses the exact geometry of $\mathcal{S}$ and does not use the tangent patch $(\boldsymbol{x}, \overrightarrow{n})$ approximation anymore,
2. the less a surface element is viewed in a given image, the less it contributes to the energy, and
3. this re-projection can easily be computed on graphics hardware with projective texture mapping.

The first point is essential to get an accurate reconstruction: in methods approximating the surface by planar patches, the choice of patch size is a difficult trade-off between robust and accurate photo-consistency.

4.4.2 Regularization

While the original intrinsic energy $\mathrm{E}_{\mathrm{photo}}$ of Equation 4.1 is self-regularizing due to the integration over the surface, this is not anymore the case of Equation 4.2. The energy function $\mathrm{E}_{\mathrm{error}}$ is thus complemented with a thin-plate energy $\mathrm{E}_{\mathrm{thin-plate}}$ which measures the total curvature of the surface. This term penalizes strong bending, not large surface area:

$$\mathrm{E}_{\mathrm{thin-plate}}(\mathcal{S}) = \int_{\mathcal{S}} \kappa_1^2 + \kappa_2^2 \, \mathrm{d}\mathcal{S} \; ,$$

where κ_1 and κ_2 are the principal curvatures of the surface at the considered point. Consequently, the associated gradient flow is exempt from the classical shrinking bias.

The total energy to be minimized by the variational refinement is finally:

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{error}}(\mathcal{S}) + \lambda_{\mathrm{thin-plate}} \, \mathrm{E}_{\mathrm{thin-plate}}(\mathcal{S}) \; ,$$

and the surface $\mathcal{S}$ follows the evolution:

$$\mathcal{S}(0) = \mathcal{S}^0 \qquad \frac{\mathrm{d}\mathcal{S}}{\mathrm{d}t} = -\nabla \mathrm{E}(\mathcal{S}) \; .$$

4.4.3 Surface representation

The level set representation used in [Faugeras and Keriven, 1998, Pons et al., 2007a] has a prohibitive computational and memory cost for high resolution reconstructions. Unstructured polygonal meshes are much better at capturing extremely fine geometry. Moreover, both the global optimization step

and the computation of the image reprojection $I_{ij}^{\mathcal{S}}$ on graphics hardware depend on a triangular mesh. Hence, the obvious choice for representing $\mathcal{S}$ is a deformable triangular mesh with vertices and triangular facets. Moreover, $\mathcal{S}^0$ is assumed to have the desired topology, which is confirmed in our numerical experiments. As a result, it was found not necessary to resort to complex remeshing procedures [Pons and Boissonnat, 2007, Zaharescu et al., 2007] to handle topology changes during deformation.

4.4.4 Discretization

An overwhelming majority of methods in variational multi-view stereovision [Faugeras and Keriven, 1998, Duan et al., 2004, Hernández and Schmitt, 2004, Jin et al., 2005, Lhuillier and Quan, 2005, Pons et al., 2007a], and more generally in computer vision, rely on an *optimize then discretize* approach: an energy functional depending on a continuous infinite-dimensional representation is considered, the gradient of this energy functional is computed analytically, then the obtained minimization flow is discretized.

In contrast, a *discretize then optimize* approach is adopted: an energy function that depends on a discrete finite-dimensional surface representation, here a triangle mesh is considered, and standard non-convex optimization tools are used. The benefits of this approach have long been recognized in mesh processing, but have seldom been demonstrated in computer vision [Slabaugh and Unal, 2005, Delaunoy et al., 2008].

Following [Delaunoy et al., 2008], the term $\mathrm{E}_{\mathrm{error}}$ from [Pons et al., 2007a] is rewritten as a function of the triangulated mesh. As recalled in [Eckstein et al., 2007, 2.2], the corresponding discrete gradient field can be directly computed with the partial derivatives of this energy with respect to vertex positions. First of all, it circumvents the difficult task of choosing a consistent discretization of differential quantities, such as normal and curvature, on a triangle mesh. Second, it is more faithful to the data, and it guarantees that the energy actually decreases: notably, the obtained gradient vector at a vertex involves integrals over the whole ring of triangular facets around it. This is in strong contrast with a point-wise, and thereby noise-sensitive, dependency on the input data that a late discretization typically causes. A crucial point has to be noted here: this discrete gradient flow may include a significant tangential component driving the vertices at the right places minimizing the energy. For instance, vertices naturally migrate to the object edges if any. This is illustrated by the crisp reconstruction of stair treads in Figures 4.7 and 4.8.

When the mesh parametrization is close to isometric, the gradient from the complementary thin-plate energy reduces to a simple bi-Laplacian Δ^2. A discrete analog of such simplified thin-plate energy and associated flow, described in [Kobbelt et al., 1998] is used by applying the umbrella operator of [Taubin, 1995] to approximate the Laplace-Beltrami operator. This

particular choice has the nice property of redistributing vertices along the surface, and in particular of discouraging degenerate triangles.

4.4.5 Balance between photo-consistency and regularization

A long-standing issue in variational methods is the proper and automatic balancing between data attachment and smoothing terms. Designing a general solution to this problem is clearly out of the scope of this chapter. A specific strategy is instead proposed that allows conducting all the following experiments *without adjusting parameters to each data set*. The solution is twofold.

First, the fact that regularization has to be more important where photo-consistency is less reliable is observed, in particular in textureless or low-textured image regions. Consequently, the contribution of camera pair (i, j) at pixel $\boldsymbol{x}_i$ in Eq. 4.2 is weighted by a reliability factor:

$$\min\{\sigma_i^2, \sigma_j^2\} / \left(\min\{\sigma_i^2, \sigma_j^2\} + \epsilon^2\right) \ ,$$

where σ_i^2 and σ_j^2 denote the local variance at $\boldsymbol{x}_i$ in images I_i and $I_{ij}^{\mathcal{S}}$, respectively, and ϵ is a constant.

Second, the two terms of the energy function are homogenized: while the data attachment term of Eq. 4.2 is homogeneous to an area in pixels, the discrete thin-plate term is homogeneous to squared world units. After weighting the contribution of each image in Eq. 4.2 by the square of the ratio between the average depth of the scene and the focal length in pixels, a scalar regularity weight can be defined whose optimal value is stable across very different datasets.

4.4.6 Mesh resolution

The resolution of the mesh is automatically and adaptively adjusted to image resolution: a triangular facet is subdivided if there exists one camera pair such that the visible facet projection exceeds a user-defined number of pixels in both images. This threshold is set to 16 pixels in the experiments. A classical one-to-four triangle subdivision scheme is used, which has the advantage of preserving sharp edges.

4.5 Experimental results

As already mentioned, the following experiments have been conducted with the same parameters. The photo-consistency h is based on the same NCC as in Chapter 1, although other more elaborate measures could be envisaged. Some operations are implemented on graphics hardware, mainly NCC estimations and image reprojections. Depending on the number of images,

the running time of our pipeline ranges from fifteen to ninety minutes, on a 3.0 GHz CPU and an NVIDIA GeForce 8800 GTX GPU. State-of-the-art results have also been obtained on the *templeRing* data set of the Middlebury challenge [Seitz et al., 2006] for small-scale objects taken under tightly controlled imaging conditions but these results are not reproduced here since the focus is on experiments on large-scale scenes.

Original data sets The method was tested on an aerial acquisition of the *Aiguille du Midi* summit (data and calibration courtesy Bernard Vallet and Marc Pierrot-Deseilligny respectively). The data set consists of 53 images of 5 Mpix. Figure 4.3 shows two of the images, the generated point cloud, the initial mesh $\mathcal{S}^0$ and the final reconstruction. This experiment validates the whole pipeline and the ability to cope with uncontrolled imaging conditions (snow, sun, moving people from one image to another) and a mix of complex and smooth geometries. The variational process is able to recover the top antenna although it is only partially present in $\mathcal{S}^0$. Figure 4.5 shows results on a data set of 27 images of 10 Mpix of a sculpted calvary taken from the ground. The cloud has $1,300,000$ points, with many outliers, mainly sky points obtained by matching clouds that have moved between shots. Only $660,000$ of these points are selected for the initial mesh $\mathcal{S}^0$ ($1,200,000$ triangles). This mesh is noisy, due to the process of matching interest points that are just approximately view-point invariant (a smoothed version of this mesh is shown in Figure 4.5). As the close views of Figure 4.6 show, the final reconstruction ($1,850,000$ triangles) is sharp enough to capture meaningful details, while global visibility is still correct.

Large-scale dense multi-view stereo benchmark Provided by Strecha *et al.* [Strecha et al., 2008], the already mentioned data sets consists in outdoor scenes acquired with 8 to 30 calibrated 6 Mpix images. Ground truth has been acquired with a LIDAR system. Evaluation of the multi-view stereo reconstructions is quantified through relative error histograms counting the percentage of the scene recovered within a range of 1 to 10 times an estimated LIDAR depth standard deviation σ. The focus here is on the four first data sets, for which other groups have submitted results. Dedicated to large-scale objects and fitting perfectly our objective, these sets are particularly challenging, especially the *castle-P19*, a complete courtyard acquired from the inside and where a tractor stays in the middle, disturbing reconstruction. So far, only Furukawa *et al.* [Furukawa and Ponce, 2007] and Tylecek *et al.* [Tylecek and Sara, 2009] submitted for these particular data sets. Figures 4.7 and 4.8 show, for the four data sets, two of the images and a global view of the generated point cloud, the initial surface and the final reconstruction. The output meshes go from $1,450,000$ to $3,000,000$ triangles, depending on the data set. Comparison with the other methods are given in Figure 4.9, where

cumulated histograms clearly show that the proposed pipeline is both more accurate (thanks to the final variational refinement) and complete (thanks the initial visibility-consistent mesh). Focusing of the *Herz-Jesu-P8* data set, Figure 4.10 provides a more visual comparison. For the four best methods so far and our method, the results are rendered and the corresponding error is color encoded. Note that the ground truth is not everywhere available (*e.g.* , the metal bar under the left porch, which we actually partially recover). Finally, Figure 4.11 compares, for the other three datasets, the rendering of our reconstruction with the one of the second best method. Further results are available on the challenge website[1].

[1] http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html

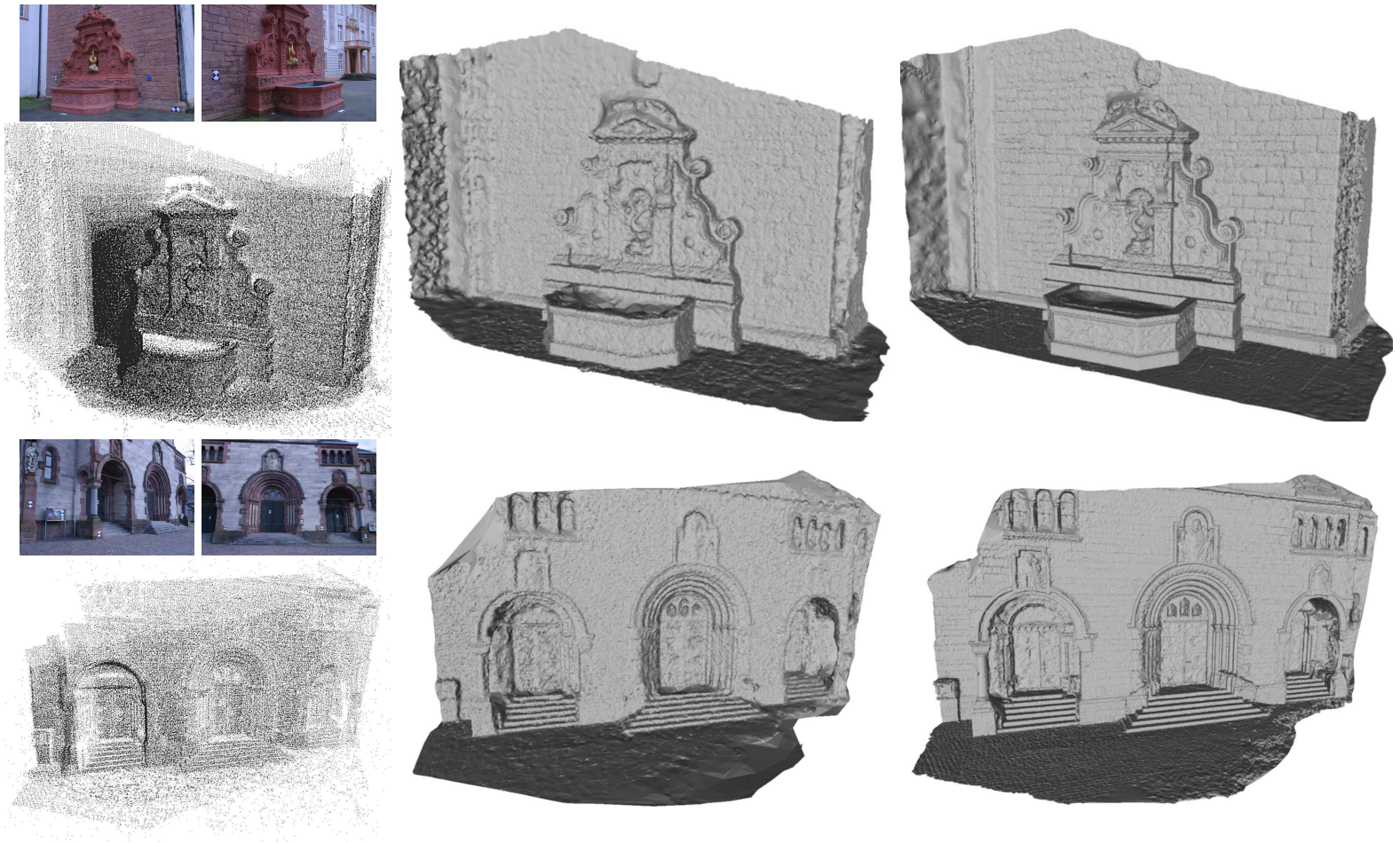


Figure 4.7: ***The four two data sets of [Strecha et al., 2008].*** *On each row, from left to right: 2 images of each data set, namely* fountain-P11 *(*11 *images) and* Herz-Jesu-P8 *(*8 *images), the generated point cloud, the initial surface (smoothed for rendering purpose) and the final reconstruction, respectively* $1,600,000$ *and* $1,450,000$ *triangles.*

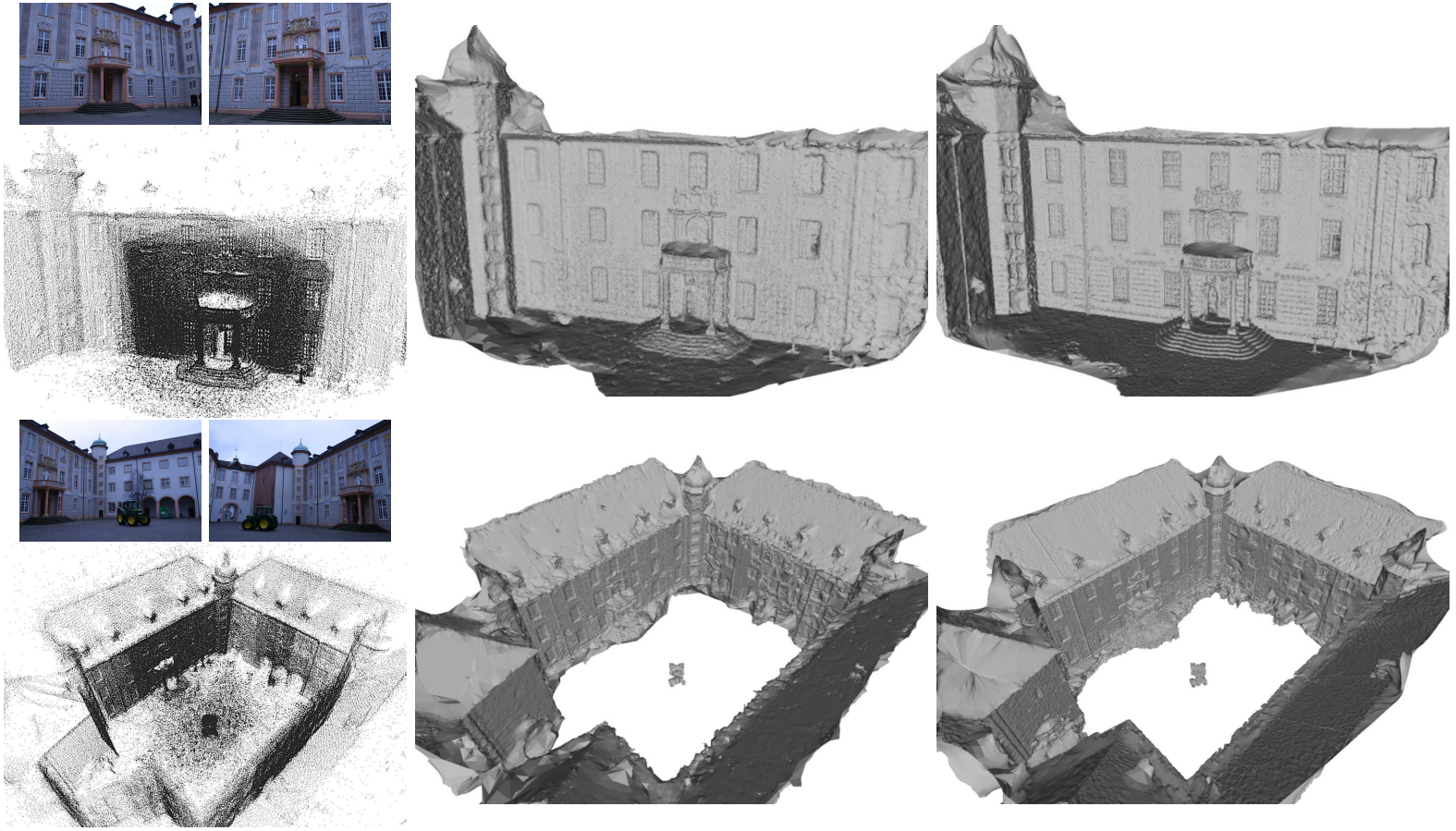


Figure 4.8: ***The next two data sets of [Strecha et al., 2008].*** *On each row, from left to right: 2 images of each data set, namely* entry-P10 *(*10 *images) and* castle-P19 *(*19 *images), the generated point cloud, the initial surface (smoothed for rendering purpose) and the final reconstruction, respectively* $2,000,000$ *and* $3,000,000$ *triangles. Note how details, topology (*e.g. *., columns) and edges (*e.g. *., stairs) are faithfully recovered while regularization still handles as correctly as possible blurred or untextured parts.*

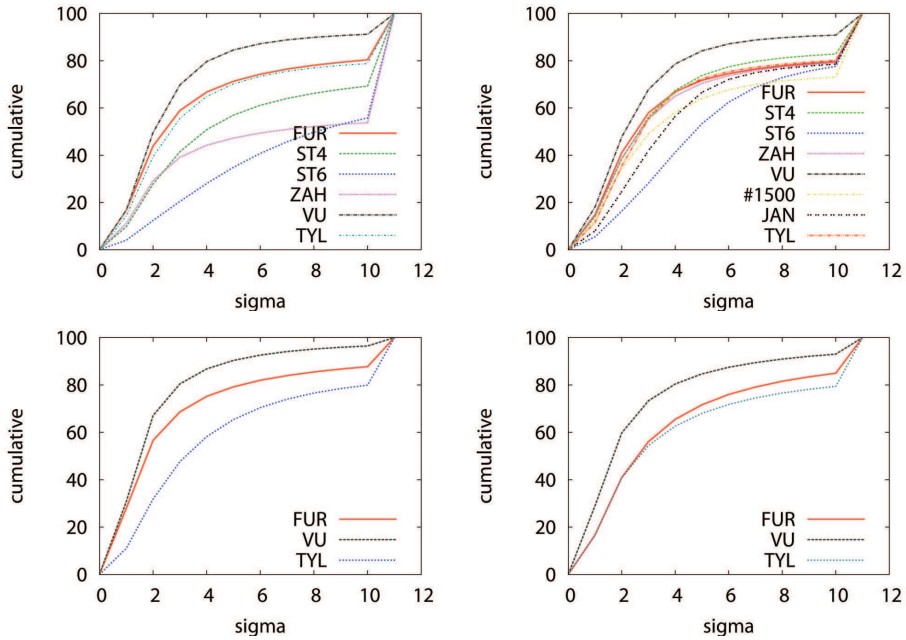


Figure 4.9: ***Relative error cumulated histograms.*** *From top to bottom, left to right, relative error cumulated histograms, respectively for the* Herz-Jesu-P8, fountain-P11, entry-P10 *and* castle-P19 *data sets. Legend is the following: FUR for [Furukawa and Ponce, 2007], ST4 for [Strecha et al., 2004], ST6 for [Strecha et al., 2006], ZAH for [Zaharescu et al., 2007], TYL for [Tylecek and Sara, 2009], JAN for [Jancosek and Pajdla, 2009], VU for our work. On all data sets, the measures confirm clearly the superiority of our better results, both in accuracy and completeness.*

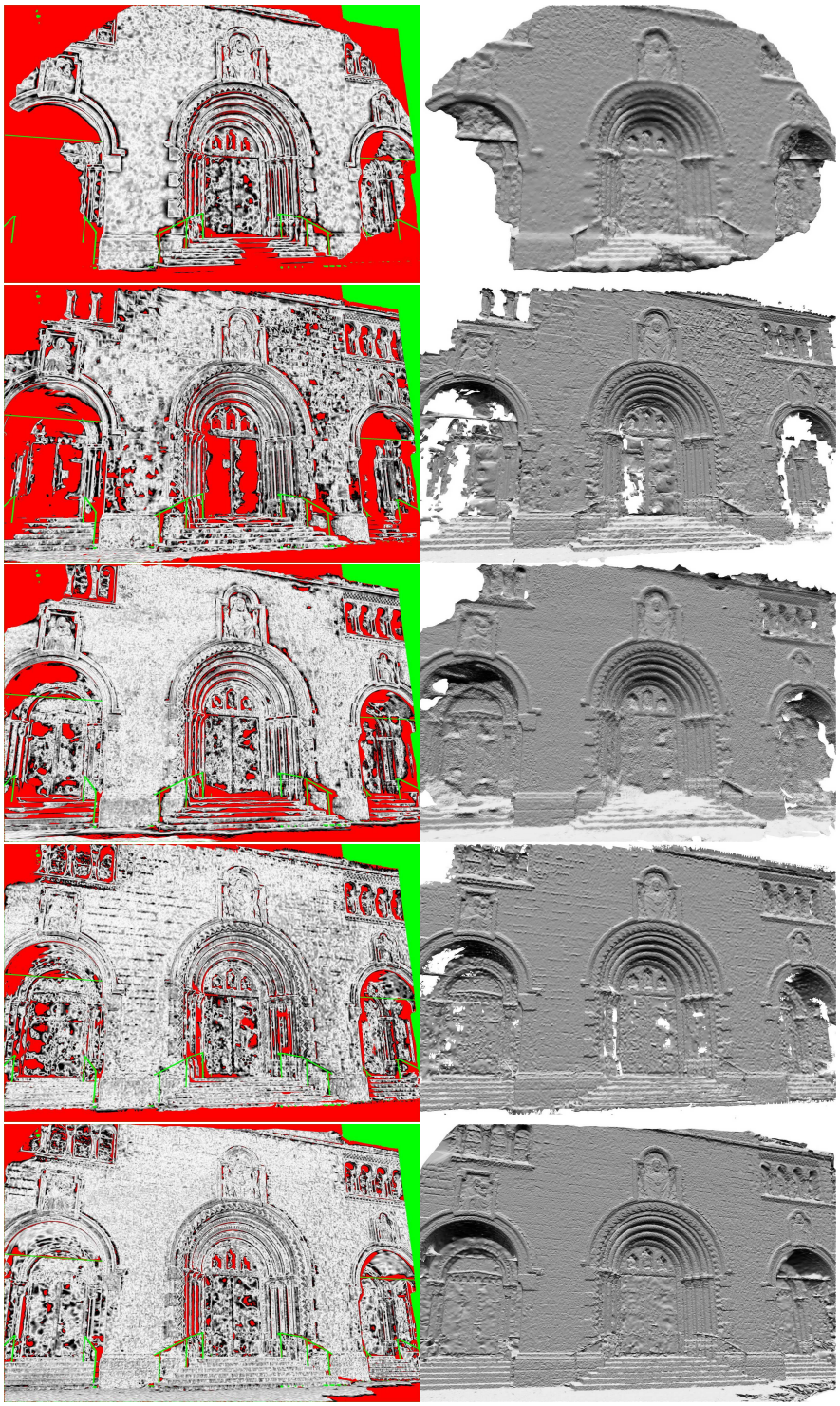


Figure 4.10: ***A visual comparison on the* Herz-Jesu-P8 *data set.*** *From top to bottom, results from ZAH [Zaharescu et al., 2007], ST4 [Strecha et al., 2004], TYL [Tylecek and Sara, 2009], FUR [Furukawa and Ponce, 2007] and our work. Left: variance weighted depth difference (red pixels encode an error larger than 3σ; green pixels encode missing LIDAR data; the relative error between 0 and 3σ is encoded in gray. Right: diffuse renderings of the corresponding triangle meshes.*

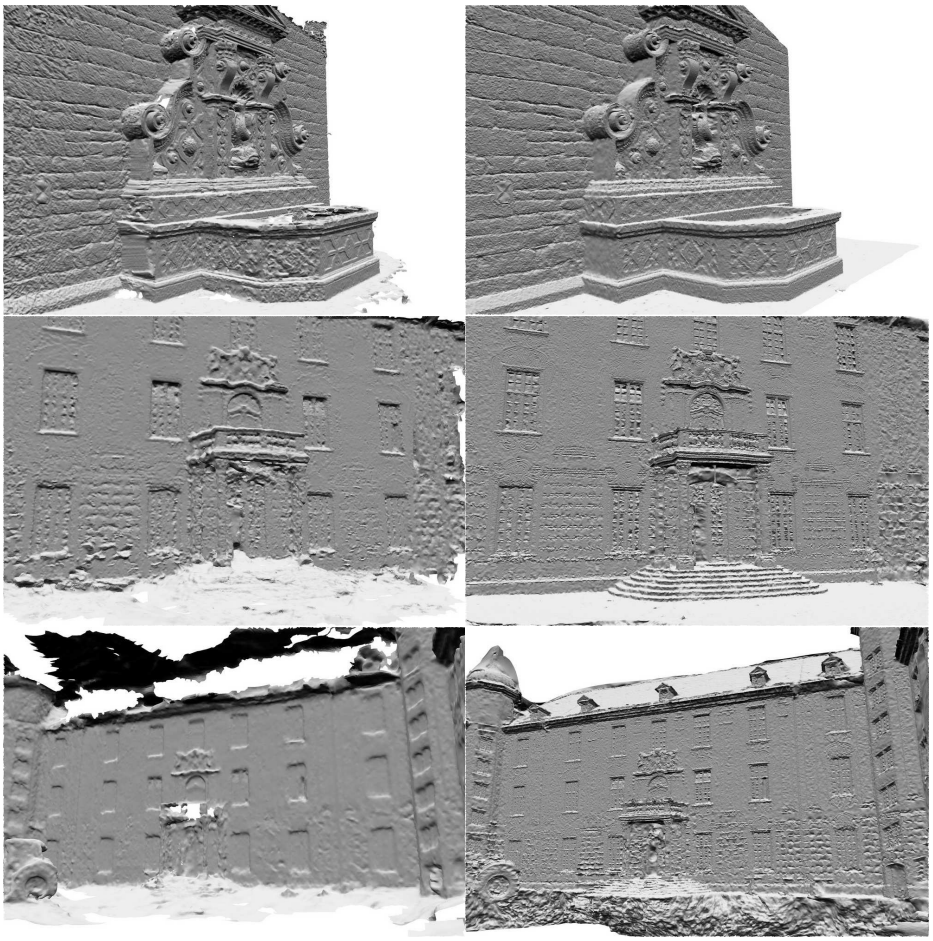


Figure 4.11: ***Visual comparisons for the three other data sets.*** *Left: rendering of the second best method ([Strecha et al., 2004] for* fountain-P11*, [Furukawa and Ponce, 2007] for the* entry-P10 *and* castle-P19*). Right: our method.*

Part II

Reconstruction of simplified models

CHAPTER 5

Proposed approach

Chapters 3 and 4 deal with methods aimed at detailed reconstructions using priors such as visibility, surface quality, fairness or photo-consistency. The current and the following chapters focus on reconstruction with strong shape priors to build geometrically simplified models from points cloud that captures the characteristic geometric features of the object or scene. A survey of related techniques shows that despite being interesting and having practical importance, this problem seems to have not been sufficiently considered. Our general approach to this problem is then introduced and advocated.

Contents

5.1 Related work

Reconstruction of 3D models from urban imagery has long been an active topic of research in computer vision and photogrammetry. Applications such as Google Earth or Microsoft Virtual Earth have allowed a broad audience to visualize large-scale models of cities with superimposed street level or aerial imagery. Still, the models are mostly handmade and automatic generation from images of such content is clearly desirable. Similarly, the automatic reconstruction of range acquired surfaces with simple primitives directly from point clouds finds strong motivations in reverse-engineering, visualization, shape indexing and recognition for instance. Several methods for automatic or interactive image-based modeling, segmentation of point clouds or surfaces have been developed, and can apply to the problem of reconstruction with shape-priors or semantic modeling. The following sections give an overview of these various attempts.

5.1.1 Dense multi-view stereo

As seen in Chapter 4, dense multi-view stereo has received a tremendous amount of attention in the last years since the comparison and benchmark of [Seitz et al., 2006]. While the accuracy of the latest results begins to challenge laser acquired data, most of these method are not suited to reconstruction in a general setting, because of their reliance on silhouette-segmented images or the necessity to tightly enclose the whole scene in a bounding-box in order to apply an optimization framework over a regular volumetric grid. If a few dense multi-view methods still seem appropriate and scalable to general outdoor scenes (such as the one described in Chapter 4), all of these approaches produce overly complex meshes (with several hundreds of thousands of triangular facets) and trade a highly detailed reconstruction for the loss of characteristic large geometric features in the scenes. The proposed approach does not suffer from any of these shortcomings and directly outputs a geometrically simple model of a potentially large-scale scene.

5.1.2 Automatic urban modeling

Dedicated methods have been elaborated for architectural scenes. The method described in [Werner and Zisserman, 2002a,b] detects dominant directions from a sparse SfM point cloud with matched lines. Vertical planes are then extracted, roof planes added and finally the ground is detected by sweeping along the vertical direction. This set of planes forms an initial coarse shell on which a restricted family of parametrized models of window, doors and dormer windows is then fitted. [Dick et al., 2004] also heavily exploits strong direction constraints but produces more elaborate buildings reconstructions thanks to a wider dictionary of parametrized models. Other approaches such

as [Cornelis et al., 2007] target the reconstruction of larger-scale city roads and coarse building facades by using road-level passive stereo on a mobile car combined with recognition. In contrast with general dense multi-view stereo methods, these approaches are limited to reconstruction of scenes where their numerous assumptions are practically verified.

5.1.3 Human-assisted image modeling

Human-assisted reconstruction was pioneered by [Debevec et al., 1996]: a user marks some edges in the images, select simple primitives, and the interface aligns the primitives with the edges (the SfM problem is part of the optimization and relies on strong architectural cues). This initial effort was a source of inspiration for the development of commercial products such as Autodesk ImageModeler [aut, 2009] which after a manual SfM step, similarly allows a user to guide a simplified model reconstruction from images by exploiting photometric cues. A similar semi-interactive modeling approach to videos has been proposed in [van den Hengel et al., 2007]. More recently, the Google Sketchup program [goo, 2009], designed to create content for Google Earth, has proposed a tool to align objects to photographs requiring the careful specification of a coordinate frame from three vanishing points (which may not always be possible for close-up pictures for instance). [Habbecke and Kobbelt, 2006] mention a prototype including an iterative plane-fitting requiring heavy user intervention to delimit matching polygonal regions in images. [Sinha et al., 2008] present an even easier approach for architectural scenes with abundant parallel lines guiding the user and constraining optimizations thanks to extracted lines and vanishing points. Apart from the prototype of [Habbecke and Kobbelt, 2006], the described methods do not require calibrated images. In our case, if such information is unavailable, it could be recovered using a combination of computer vision techniques as noted in Chapter 1.

5.1.4 Towards automatic compact modeling from images

Very few work has tried to solve the suggested problem from images. [Fraundorfer et al., 2006] matches affine-invariant features to initialize planar seed regions which are then expanded with a region growing scheme. [Bartoli, 2007] modifies the traditional RANSAC technique [Fischler and Bolles, 1981] to facilitate multiple structures detection and subsequently computes delineation of the extracted planar regions. These two approaches focus mostly on the plane extraction step applied to sparse point clouds from SfM, extract a limited number of planes and are not designed to output simplified piecewise-planar dense reconstructions. In [Gallup et al., 2007], dominant directions are extracted from a SfM point cloud using minimum-entropy histograms. Multiple plane sweeps are then performed along these directions to

generate a set a depth hypotheses for each pixel. A best cost or an optional graph cut optimization step finally generates piecewise-planar depth maps. The method is designed for ground-level acquisitions and assumes the existence of a ground plane and perpendicular facade normals. The very recent work of [Furukawa et al., 2009] follows the same approach. It is restricted to scenes satisfying the Manhattan hypothesis [Coughlan and Yuille, 2003] and featuring planes aligned with three principal orthogonal directions. After identifying three approximately orthogonal axes with a normal histogram, plane candidates are extracted with a mean shift clustering along these three directions, after which piecewise-planar depth maps are also generated with a graph cut labeling. The proposed method heavily relies on the dense filtered and refined point clouds of [Furukawa and Ponce, 2007] and the 3 directions assumption limits its application. Moreover, no structure preserving surface reconstruction complements the method.

On the other hand, [Hilton, 2005] focuses on the mesh reconstruction and proposes a visibility-consistent construction using a constrained Delaunay triangulation of a sparse 3D features from SfM. Their approach reconstructs compact triangulated models but is unable to identify large planar polygonal facets. The method is moreover interpolatory and in practice, quite limited in robustness to outliers. The most related work is perhaps [Baillard and Zisserman, 1999] which describes a piecewise-planar reconstruction method exploiting detection of half-planes supported by matched extracted edges. The mesh reconstruction relies on less robust heuristic grouping rules. Furthermore, the technique only seems applicable to scenes mostly taken from very similar points of views (roof structure from aerial views for instance).

5.1.5 Segmentation of range data

A lot of work has been devoted to the segmentation of range images in primitives (a classification and evaluation is presented in [Hoover et al., 1996]): these approaches focuses on 2.5D data structured as depth images. As noted in Chapter 1, this kind of data is known to exhibit much lower noise levels than point clouds from passive stereo, are much denser, have a more uniform sampling and contain very few outliers (even so that high order differential quantities as curvatures can be reliably estimated and used to classify the points/pixels). Few authors have considered the problem of segmenting a whole 3D point cloud. In [Gelfand and Guibas, 2004], the notion of slippage motion is defined as a rigid motion that slides a transformed shape against its stationary version without forming gaps. Slippable components of data (including shapes such as planes, spheres and cylinders) are discovered and aggregated into regions. The approach applies to range data only without actual surface reconstruction. More recently, the RANSAC robust regression is applied in [Schnabel et al., 2007] with a focus on performance to segment point clouds acquired with 3D range scanning: several modifications make

the method scalable to relatively large set of scans with a very reasonable running time. The method relies on correct and consistent estimated normals for detection and is thus sensitive to large amounts of noise. It also assumes a dense and uniformly sampled point cloud which limits it to range data. This work is used in [Schnabel et al., 2009] for surface completion from point clouds with primitive shapes: the volumetric surface reconstruction algorithm derives from [Lempitsky and Boykov, 2007] but is guided by the recovered shapes and a connectivity criterion. Unfortunately, one of the added energy terms forces the authors to replace the minimum *s-t* cut optimization with an iterative greedy minimization whose convergence is not guaranteed. The final reconstructed surface is extracted from a dense voxel grid with [Kobbelt et al., 2001] and a number of specific variations. Finally, in [Chen and Chen, 2008], a semi-automatic reconstruction method from range scans of architectural scenes is proposed. The point set is first segmented into planes by using the Gauss map, tentative boundaries of the planar region are extracted and a final polyhedral model is built with some user interaction. The method is demonstrated on two scanned buildings but again relies heavily on correct normal estimation, user interaction and numerous heuristics.

5.1.6 Geometry processing for simplification

One may argue that standard geometry simplification techniques already exist and could straightforwardly be applied to the output meshes of a dense multi-view stereo pipeline such as the one presented in Chapter 4. However such combination is not only less efficient than the approach we propose, it is also less powerful and not adequate. First, mesh simplification (a reference on this topic is [Luebke et al., 2002]) is mainly suited to perfectly meshed, almost noise-free surfaces far from the typical output of multi-view reconstruction algorithms. These methods mainly use an edge collapse operation on triangular meshes which merges the two incident vertices of an edge into a single vertex that is optionally moved to a new location. The edge to collapse is selected via a priority queue which contains valid candidates sorted by some criterion so as to minimize the distortion error. In this spirit, the *Progressive Meshes* of [Hoppe, 1996] generates level of details from an initial mesh always using the initial mesh as a reference. The criterion is then modified to allow view-dependent simplification in [Hoppe, 1997]. [Garland and Heckbert, 1997] defines the *Quadric Error Metric* (QEM) that measures how well a new vertex would fit the supporting planes of the triangles adjacent to the selected collapsed edge. In [Hoppe, 1999] QEM is extended to account for attributes attached to the mesh. [Lindstrom and Turk, 1998] proposes to select edges and place new vertices in order to minimize the change in volume of the solid bounded by the mesh. In [Lindstrom and Turk, 2000], the simplification criterion comes from rendered images of the original, detailed mesh.

Despite these numerous variants, sometimes close to image-based compact modeling, the simplified meshes still inherit from the original mesh through a series of local modifications only, the approach does not allow to capture more global shapes without significant simplification of the mesh and curved geometry is often lost. Simplification also often requires user intervention for quality inspection. Figure 5.1 shows such an attempt to simplify a mesh of a high-quality multi-view stereo reconstruction described in Chapter 4.

(a) Reconstruction result of the *Herz-Jesu-P25* data set from the dense multi-view stereo pipeline of Chapter 4.

(b) Reconstruction result, 128× simplified, numerous details are still present.

(c) Reconstruction result, 4096× simplified, the ground and facade are planar, but the columns and the stair treads have been wrecked.

Figure 5.1: ***Various levels of simplification*** *of the output of the dense multi-view stereo pipeline with the mesh simplification scheme of [Garland and Heckbert, 1997].*

5.1.7 Geometry processing for shape recovery

Recently, more involved mesh segmentation methods have appeared and produced impressive results as a preliminary step to guide subsequent simplification or remeshing (a recent and complete survey is presented in [Shamir, 2008]). A typical example of such method is the variational shape approximation (VSA) framework of [Cohen-Steiner et al., 2004]. VSA computes an optimal partition of a mesh in k regions $\mathcal{R} = \{\mathcal{R}_i\}$ (each region $\mathcal{R}_i$ is a union of facets) represented by corresponding shape proxies $P = \{P_i\}$ (planes in the original method) so that the total distortion $E(\mathcal{R}, P) = \sum_{i=1,\ldots,k} E(\mathcal{R}_i, P_i)$ is minimized. The metric measuring region distortion is either the standard $\mathcal{L}^2$ metric between the region $\mathcal{R}_i$ and its plane proxy P_i or a new $\mathcal{L}^{2,1}$ metric, which is basically a $\mathcal{L}^2$ measure of the normal field and can be computed in closed form for plane proxies. To find an optimal partitioning, a variant of Lloyd's algorithm [Lloyd, 1982] (the algorithm at the core of k-means clustering [Steinhaus, 1956]) is used each time an additional region is introduced. The method is not fully automatic and also has to rely on heuristics (like "teleportation" of proxies) to avoid local minima but has been successfully demonstrated for coarse remeshing. Since this initial work, a number of variants have been developed by other authors, generalizing the approach to other classes of shapes to obtain better approximations with less prim-

itives: [Wu and Kobbelt, 2005] extends the methods to spheres, cylinders and rolling-ball blend patches by using approximated $\mathcal{L}^2$ and $\mathcal{L}^{2,1}$ metric and [Yan et al., 2006] proposed to approximate the original $\mathcal{L}^2$ metric to extract general quadrics along with a graph cut method to smooth irregular boundary curves.

It must be noted that these techniques face similar problems as simplification methods with imperfect inputs. Moreover, as in mesh simplification, the original, given mesh is constantly used as a reference to measure distortion and to extract the connectivity of the final approximant. Our method tries addressing a more challenging problem. Starting from the point cloud, it implicitly combines mesh segmentation and shape-based simplification during surface reconstruction. The acquisition process is accounted for throughout the pipeline while the mentioned post-processings are more likely to worsen initial errors produced by surface reconstruction algorithms.

In light of the previous analysis, we draw the conclusion that no satisfying, general enough method exists to automatically build compact shape-based models of scenes from images or point clouds. Applications include not only image-based compact modeling or compact surface reconstruction from point clouds, but also reverse-engineering or object and scene recognition, interpretation and indexing and more generally semantic modeling from images or point sets.

5.2 Overview of the proposed approach

Since the goal is to reconstruct geometrically simple models that capture the main geometric features of an object or a scene, we propose to adopt the following approach:

1. Initially and as in the previous chapters, a point cloud sampling the scene or the object of interest is acquired either through range measurements or quasi-dense passive stereo (see Figure 5.2(a));
2. From this noisy point cloud potentially containing outliers, shapes from predefined classes of shapes are robustly extracted and the points are clustered (see Figure 5.2(b)). A corresponding hierarchical description (Figure 5.2(c)) of the scene or object is built by mixing the shape detection with the induced spatial subdivision (see Figure 5.2(d));
3. The actual partition of space induced by this scene description is then explicitly computed, in particular the adjacency graph of the cells of the corresponding complex (see Figure 5.2(e)) is constructed. By labeling the cells of this subdivision as inside or outside of the scene, a mesh representing the scene can be extracted as a union of facets of the cell complex (see Figure 5.2(f)).

The spatial partitioning and subdivision of space we use is a generalization of binary space partitioning trees (BSP tree for short). Originally developed to address the hidden-surface problem [Fuchs et al., 1980], the BSP tree is a versatile structure widely used both for spatial partitioning and boundary representation (of solid objects) with particular applications in rendering, robot motion and path planning. A BSP tree is a binary tree that defines a recursive partition of space into pairs of subspaces with respect to dividing planes at arbitrary positions and orientations. Instead of being limited to planes only, any oriented hypersurface can also be used to split the space into two halves: one negative and one positive. Each node of the binary tree corresponds to a splitting hypersurface and each leaf to an unpartitioned area of space. The generalized shape-based BSP tree is first used as a data partitioning structure to hierarchically cluster a point cloud into sets of shapes and then serves as a boundary representation for surface reconstruction.

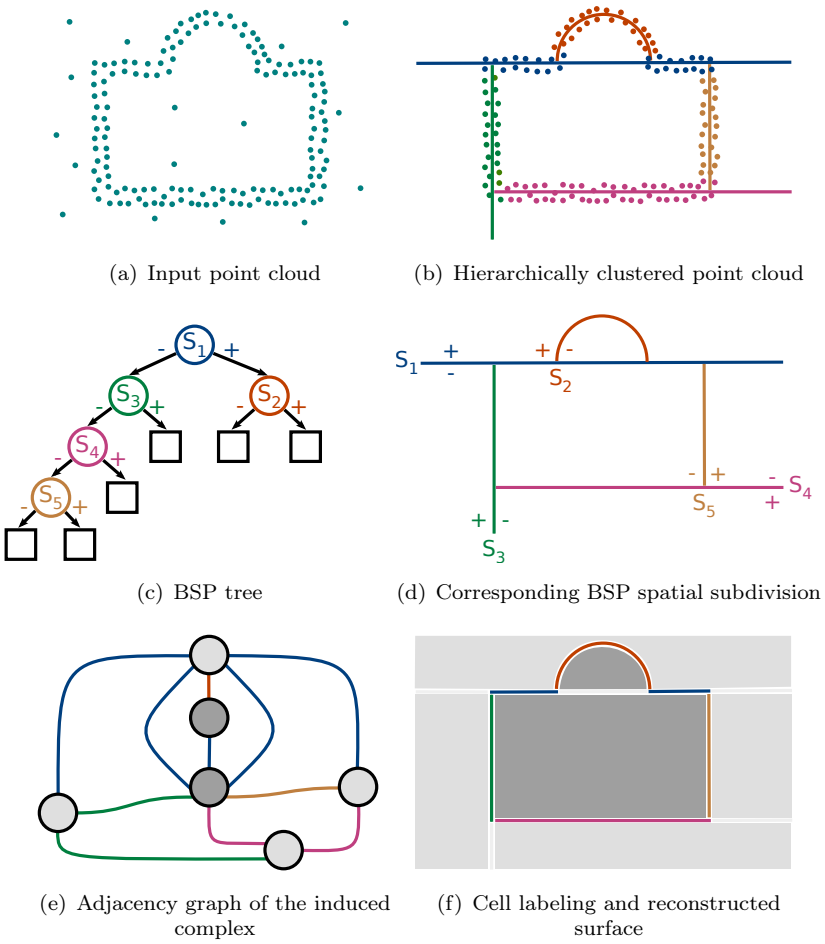
For the final cell labeling step, a variant of the visibility-based surface fitting framework of Chapter 2 is again applied. The energy defined is suitable for minimum *s-t* cuts optimization allowing a globally optimal labeling of cells and a corresponding surface to be generated from the space subdivision and the fitted points. In Chapter 8, the final step is also extended to output partly shape-based reconstructions (hybrid reconstructions) combining shape elements in some areas with non-shape based parts.

Several reasons motivates this top-down approach:

- while the initial shape extraction is greedy, it is based on a variant of the RANSAC framework more robust and suited to dense enough point clouds with multiple structures. This variant is applicable to point clouds of lower quality typical from our passive stereo techniques,
- shapes with infinite extent can naturally cope with areas of poor texture in image-based modeling or varying sampling density in range scanning,
- the detection of potential large shapes early in the reconstruction pipeline allows these geometric features to be correctly identified and recovered in the reconstruction which would be difficult if a generic surface reconstruction algorithm was first applied as in Chapter 4.

As regards the use of a generalized BSP tree as a spatial subdivision and a boundary representation guided by recovered shapes, our particular choice has a number of desirable properties for the considered problem:

- exploiting the space subdivision during the detection helps reducing the generated shape hypotheses to more useful hypotheses instead of repetitively sampling the whole point cloud,
- the use of a binary space partition instead of an arrangement limits the combinatorial complexity of the cell complex *w.r.t.* the number of recovered shapes (see Appendix B) and the corresponding cells and



(a) Input point cloud (b) Hierarchically clustered point cloud

(c) BSP tree (d) Corresponding BSP spatial subdivision

(e) Adjacency graph of the induced complex (f) Cell labeling and reconstructed surface

Figure 5.2: *Overview of our approach to shape-based reconstruction.*

facets are also more relevant for the final reconstruction,

- the connectivity of primitives and the recovery of boundaries and vertices between the different primitives, one of the major difficulty of the problem, is handled implicitly in the surface reconstruction,
- poorly point sampled areas (due to low texture in stereo) are automatically extended to meet other primitives,
- wrongly detected shapes do not significantly affect the spatial subdivision as they only split facets and cells,
- the final segmentation is implicit to the surface reconstruction and the output of our reconstruction pipeline is a piecewise-primitive segmented mesh. This segmentation may possibly be used for several purposes: remeshing, guided simplification or higher level applications like shape matching, indexing, ...

5.3 Conclusion

After a survey of applicable approaches to simplified modeling from images or range clouds, our overall approach to simplified modeling has been presented and motivated. The next chapter details the robust hierarchical shape extraction. The two subsequent chapters first describe the reconstruction of concise models when the set of shapes is restricted to planes and then the reconstruction of simplified models from more complex shape classes.

CHAPTER 6

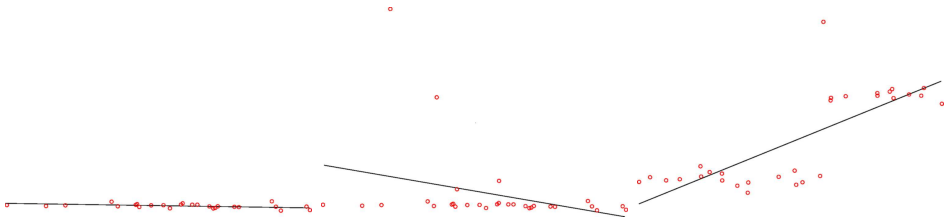
Shape extraction

As described in Chapter 5, the first step in our approach to shape-based surface construction consists in extracting shapes from the point clouds using a predetermined set of shape classes. First, the known limitations of robust regression methods for multiple shape fitting are pointed out. Then our single shape extraction based on the random sampling framework is presented and used to compute a hierarchical decomposition of the input point cloud.

Contents

6.1 Robust regression for shape fitting

Point clouds from range data or passive stereo are often more or less noisy but may also contain significant amounts of outliers as seen in Chapter 1. Furthermore, the object or scene of interest is usually complex and explainable with several different structures, which may make the estimation of a single shape even more challenging. As shown in Figure 6.1, standard regression fails in the presence of outliers, *i.e.* , small or large departures from the assumed model (in the figure, measurements are assumed to lie on a line and to be corrupted with Gaussian noise). Robust methods that are tolerant to such measurements are thus required to detect the different structures sampled by a point cloud. A large body of work has been dedicated to robust regression. Most techniques widely used in computer vision (where the issue of robustness, for instance to mismatches, is important) can be seen as optimizing some objective function (as we will see, the number of bin votes for the Hough transform and the number of outliers outside a band for RANSAC) with an appropriate sampling of the model parameters (either by quantizing the parameters space or by randomly sampling models supported by sets of points). Before describing our approach to robust regression in our specific case of dense or quasi-dense point clouds, we give an overview of the general possible approaches and their various limitations when multiple shapes have to be fitted to a noisy point cloud with numerous potential outliers.



(a) A line fitted to a set of points with (total) least-squares

(b) A line fitted to the same set of points corrupted with a few outliers

(c) A line fitted to a set of points with two different structures

Figure 6.1: ***Outliers and pseudo-outliers*** *can wreck the least-squares estimate of a line.*

In the sequel, we assume N measurements are given as a set of points $\mathcal{P} = \{P_1, \ldots, P_N\}$ in $\mathbb{E}^d$, these measurements may be noisy and carry outliers, *i.e.* , points that do not sample the real object or scene. To fit a shape $\mathcal{Sh}(\boldsymbol{\beta})$ parametrized by n parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)^T \in \mathbb{R}^n$ ($n = 4$ for a plane, $n = 4$ for a sphere, etc. . .), the residual of each measurement $r_i(\boldsymbol{\beta})$ is considered. This residual usually takes the simplest form of some distance measure to the parameterized shape $\mathcal{Sh}(\boldsymbol{\beta})$: $r_i(\boldsymbol{\beta}) = d(P_i, \mathcal{Sh}(\boldsymbol{\beta}))$.

6.1.1 M-estimators

M-estimators (M for "maximum likelihood type") [Huber, 1981] generalize maximum likelihood estimation by weighting differently each measurements according to some robust loss function ρ and estimate a model $\boldsymbol{\beta}^\star$ as follows:

$$\boldsymbol{\beta}^\star = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \rho\left(\frac{r_i(\boldsymbol{\beta})}{s\sigma}\right) ,$$

where ρ is the loss function, often taken as:

$$\rho(x) = \begin{cases} 1 - \left(1 - x^2\right)^3 & |x| \leq 1 \\ 1 & |x| > 1 \end{cases} ,$$

which effectively limits the influence of outliers. σ is the assumed noise level and s a (user-specified) scale parameter. For most choice of ρ there is no closed-form solution, and the optimization can be reduced to "iteratively reweighted least-squares" requiring a good initialization. This initialization may restrict the use of M-estimator and often has to come from other robust methods to avoid leverage measurements that may lock to M-estimator optimization in a local minimum.

6.1.2 Finding modes in parameter space

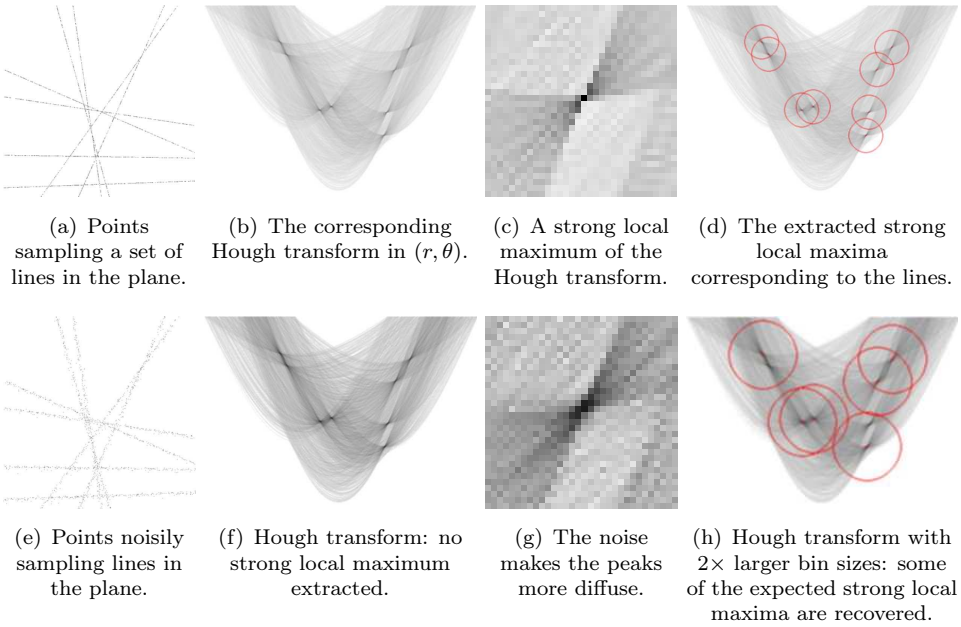
A general approach to shape detection consists in mapping the input data into the parameter space and detect peaks in the corresponding distribution of shapes. These techniques can generally handle very naturally the detection of multiple structures. The classical Hough transform is first described and then the variant of the general mean-shift procedure that applies in our context is presented.

Hough transform and variants

The Hough Transform [Hough, 1959, 1962, Duda and Hart, 1972] (SHT for Standard Hough Transform) is a widely used technique to extract shapes that considers the number of measurements belonging to each hypothesized shapes in a discretized version of the parameter space of the shapes. Noting $\boldsymbol{\beta}_{j_1,\ldots,j_n}$ the discretized parameters:

$$\boldsymbol{\beta}^\star = \arg\max_{\boldsymbol{\beta}_{j_1,\ldots,j_n}} \left| \left\{ i \in [\![1, N]\!] \ : \ r_i^2(\boldsymbol{\beta}_{j_1,\ldots,j_n}) \leq r^2_{\max j_1,\ldots,j_n} \right\} \right| ,$$

where $r_{\max}$ is a threshold depending on the discretization and the chosen distance to shapes. Although this optimization is performed over a shape domain with discretized parameters, this formulation exhibits strong similarities with M-estimators:



(a) Points sampling a set of lines in the plane.

(b) The corresponding Hough transform in (r, θ).

(c) A strong local maximum of the Hough transform.

(d) The extracted strong local maxima corresponding to the lines.

(e) Points noisily sampling lines in the plane.

(f) Hough transform: no strong local maximum extracted.

(g) The noise makes the peaks more diffuse.

(h) Hough transform with $2\times$ larger bin sizes: some of the expected strong local maxima are recovered.

Figure 6.2: *The Hough transform has difficulties handling noise.*

$$\boldsymbol{\beta}^\star = \underset{\boldsymbol{\beta}_{j_1,\dots,j_n}}{\arg\min} \sum_{i=1}^{N} \rho\left(\frac{r_i(\boldsymbol{\beta}_{j_1,\dots,j_n})}{r_{\max j_1,\dots,j_n}}\right) ,$$

where $\rho(x) = \begin{cases} 0 & |x| \leq 1 \\ 1 & |x| > 1 \end{cases}$.

In an actual implementation, the parameter space is first discretized and each measurement votes in all the bins of shape hypotheses to which it may belong (up to some tolerance). The optimal hypothesis is the one whose bin has accumulated the most votes. To extract multiple structures, different strategies are possible, such as iteratively extracting the global maximum, removing its contribution from the shapes histogram and iterating until either the desired number of shapes is met or the maximum is not strong enough. Another option consists in extracting at once all strong enough local maxima and keeping only the strongests. While the Hough transform naturally handles data with multiple structures (of the same kind), the bin size and threshold adjustment are delicate with a compromise between accuracy and computational efficiency and the method faces inherent difficulties on noisy data.

We give an example in the simple case of lines detection in the plane. Lines can be parametrized as follows: $\boldsymbol{\beta} = (r, \theta)$ and a point of coordinates (x, y) in the plane belongs to all the lines whose parameters obey the following relation:

$$r(\theta) = x\cos\theta + y\sin\theta = r_0\cos(\theta - \theta_0) \ .$$

with $r_0 = \sqrt{x^2+y^2}$ and $\theta_0 = \cos^{-1} x/r_0 = \sin^{-1} y/r_0$. This relation describes a sinusoid in Hough space. Figure. 6.2 illustrates the results of a successful extraction of lines and the difficult choice of an adequate bin size when noise is present: the peaks are more diffuse and missed by the detection. While augmenting the bin size can help recovering some of the missing peaks some are definitely lost. The fact that the size of the voting space is exponential in the number of parameters is another problem and makes SHT impractical for many applications involving detection of complex shapes.

Several variants of the SHT have been tried to address to some of these issues. The Probabilistic Hough Transform [Kiryati et al., 1991] reduces the number of votes by randomly selecting a fraction of the input points while maintaining results similar to the SHT. The Randomized Hough Transform (RHT) [Xu et al., 1990] only generates line hypotheses for pairs of points simplifying the voting and revealing shapes more quickly. The Hierarchical Hough Transform [Princen et al., 1990] iteratively considers coarse-to-fine versions of the feature space. In contrast, the Adaptive Hough Transform [Illingworth and Kittler, 1989] guides the refinement of both the range and number of bins of the discretization of the parameter space. Finally, while the SHT applies to any shape with an analytic equation (*e.g.* line, circle, etc...), it is limited to 3 parameters for computational efficiency reasons, and schemes to split the search in lower dimensional parameters subspaces have to be used. The Generalized Hough Transform [Ballard, 1981] is able to detect arbitrary objects by searching for transformation parameters instead. Finally, a recent applicative variant of the SHT can be found in [Décoret et al., 2003]: the extreme simplification of models into a set of partially transparent texture-mapped polygons is done with a different objective function which combines a facet coverage and a penalty cost per plane.

Mean-shift derivatives

Instead of quantizing the parameter space and accumulating votes, the data can be mapped into the parameter space and the mean-shift procedure of [Comaniciu and Meer, 2002] used to find the maxima of an estimated density. Given N data points $\boldsymbol{x}_i \in \mathbb{R}^d$, the kernel density estimate (also named Parzen window method [Parzen, 1962]) can be applied to give a non-parametric estimator $\hat{f}_k$ of the density at $\boldsymbol{x}$:

$$\hat{f}_K(\boldsymbol{x}) = \frac{c_{K,h}}{N}\sum_{i=1}^{N} K\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right) \ ,$$

where K is a *kernel function* satisfying $K(\boldsymbol{x}) \geq 0$, h is a smoothing parameter called the *bandwidth*, and the constant $c_{K,h}$ is chosen to ensure that $K(\boldsymbol{x})$

integrates to 1. The kernel used in practice are radially-symmetric, $K(\boldsymbol{x}) = k(\|\boldsymbol{x}\|^2)$ where $k(x)$ is a *profile* defined for $x \geq 0$. A common choice of profile is $k(x) = \exp(-x/2)$ yielding a multivariate normal kernel K. The bandwidth h globally controls the width of the kernel placed at each x_i. Define $g(x) = -k'(x)$ and $G(\boldsymbol{x}) = c_{G,h} g(\|\boldsymbol{x}\|^2)$ the associated kernel with normalization constant $c_{G,h}$. Taking the gradient of this density estimate, it can be shown that:

$$\boldsymbol{m}_h(\boldsymbol{x}) = C \frac{\nabla \hat{f}_h(\boldsymbol{x})}{\hat{f}(\boldsymbol{x})} = \frac{\sum_{i=1}^N g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right) \boldsymbol{x}_i}{\sum_{i=1}^N g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right)} - \boldsymbol{x} \ ,$$

where C is a positive constant and $\boldsymbol{m}_h(x)$ is called the *mean shift* vector. The expression shows that the mean shift vector is proportional to the normalized density gradient estimate at x. The iteration:

$$\boldsymbol{x}^{j+1} = \boldsymbol{x}^j + \boldsymbol{m}_h\left(\boldsymbol{x}^j\right)$$

is a gradient ascent technique converging to a stationary point of the density. Saddle points can be detected and removed to obtain only the modes of $\hat{f}(\boldsymbol{x})$. As shown in [Comaniciu et al., 2001], the mean-shift procedure can be made more adaptive by introducing one bandwidth parameter h_i and one weight w_i per point. Setting these is again problem-specific, but a typical heuristic consists in computing the k nearest neighbor $\boldsymbol{x}_{i,k}$ of each point $\boldsymbol{x}_i$ (for a fixed k) to set the bandwidth based on some distance measure between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i,k}$.

The mean-shift procedure is a non-parametric estimator and is not directly applicable to fitting shapes. The data has first to be mapped into some parameter space. The projection-based mean-shift (pbM) of [Subbarao and Meer, 2009] proposes to reformulate the M-estimator objective function as a variable bandwidth kernel density estimation to fit hyperplanes in $\mathbb{R}^d$. The method turns into a projection pursuit approach by seeking modes with mean-shift along directions hypothesized through random sampling. However the technique is restricted to plane detection and is not intrinsically robust. In addition, and as for the Hough transform, the approach is limited in robustness *w.r.t.* the choice of parametrization.

Generally the two parameter space methods described are unable to cope with large percentage of gross outliers, especially with a growing number of structures and when the distribution of inliers per structure becomes uneven.

6.1.3 Random sampling optimization

Two well-known objective functions in robust regression rely on random sampling optimization.

Least median of squares (LMS)

LMS [Rousseeuw, 1984] estimates an optimal model $\boldsymbol{\beta}^\star$ that yields the smallest value for the median of the squared residuals computed for the entire data set:

$$\boldsymbol{\beta}^\star = \arg\min_{\boldsymbol{\beta}} \operatorname*{median}_i r_i^2(\boldsymbol{\beta}) \ .$$

The method is very robust to outliers and does not need a user-specified scale estimate. A robust scale estimate can actually be inferred from the optimal median:

$$\sigma^\star = C\sqrt{\operatorname*{median}_i r_i^2(\boldsymbol{\beta})} \ ,$$

where C is a constant making $\sigma^\star$ unbiased for a target distribution of measurements errors. Inliers to the optimal model are identified to have a residual within some multiple of $\sigma^\star$ (or geometrically lying within a band defined by a multiple of $\sigma^\star$ around the model). Note that the median of an unordered set can be found in expected linear time only using the QuickSelect algorithm.

A first drawback limits the applicability of LMS to the data sets we consider where other structures act as outliers during one regression: its breakdown point is of 0.5 meaning that whenever the number of outliers exceed half of the size of the data set, the estimate is false. A second drawback is its poor efficiency meaning that the scale is usually underestimated under a Gaussian noise error model [Rousseeuw and Leroy, 1987].

The MINPRAN estimator of [Stewart, 1995] (minimize probability of randomness) is an extension of LMS that tolerates more than 50% outliers but assumes an explicit outlier distribution. Least trimmed squares improve the breakdown point of LMS by replacing the median with another single k-th order statistics. Other extensions [Miller and Stewart, 1996, Lee et al., 1998] have been proposed to compute several k-th order statistics of the squared residuals instead. In [Lee et al., 1998], the value of k is determined adaptively from the data and in [Miller and Stewart, 1996] unbiased scale estimates are used to find the smallest scale over all possible k. Both methods have been demonstrated for range image segmentations but their requirement to systematically estimate large numbers of order statistics (and thus to presort the residuals) may however limit a practical application to large point sets.

Random sampling consensus (RANSAC)

RANSAC [Fischler and Bolles, 1981], a method that has known a great success in computer vision, can cope with large proportion of outliers and has found numerous practical applications. RANSAC maximizes the number

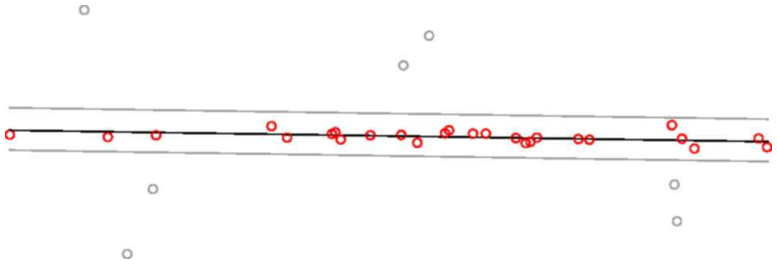


Figure 6.3: ***The RANSAC objective function*** *counts the number of points "within a band around" a hypothesized model.*

of inliers, defined as having a squared residual lower than some user-specified threshold:

$$\boldsymbol{\beta}^\star = \arg\max_{\boldsymbol{\beta}} \left| \left\{ i \in [\![1, N]\!] \ : \ r_i^2(\boldsymbol{\beta}) < r_{\max}^2 \right\} \right| \ .$$

Equivalently, RANSAC minimizes the number of outliers outside a band induced by this threshold (see Figure 6.3) and can again be reformulated in a way similar to M-estimator:

$$\boldsymbol{\beta}^\star = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \rho\left(\frac{r_i(\boldsymbol{\beta})}{r_{\max}} \right) \ ,$$

with $\rho(x) = \begin{cases} 0 & |x| \leq 1 \\ 1 & |x| > 1 \end{cases}$. While the objective function bears some similarity with that of the Hough Transform, the optimization is done over shapes hypothesized in the space of measurements and the threshold $r_{\max}$ has a clear interpretation *w.r.t.* these measurements.

MSAC [Torr and Zisserman, 2000] replaces the binary loss function of RANSAC with $\rho(x) = \begin{cases} x^2 & |x| \leq 1 \\ 1 & |x| > 1 \end{cases}$ to also penalizes inliers according to their residuals. While a large number of RANSAC derivatives have been developed, RANSAC is biased and has known limitations [Stewart, 1997] when applied to data featuring multiple structures and small-scale discontinuities.

Random sampling procedure

Since the objective functions of LMS and RANSAC described are not differentiable, an alternate optimization procedure must be devised. A possible form of a random sampling optimization is shown in Algorithm 2. A number of model hypotheses are randomly generated from minimal sets of measurements and the objective function is evaluated on each model. If the model is judged satisfactory (with a finite error), then it is compared to the current

optimal model which is possibly updated. When the generation of hypotheses is over, the best model encountered so far is kept along with a set of inliers and the optimal value of the objective function. This best model is often "polished" and re-estimated from the set of inliers.

Algorithm 2 Optimization through random sampling to fit a parametrized model $\mathcal{SH}(\boldsymbol{\beta})$ from a set of samples.

function RANDOMSAMPLING($\mathcal{P}$, ...)
  $i \leftarrow 0$ ▷ the number of drawn models
  $k \leftarrow k_{\min}$ ▷ the min. # of models to draw
  $\mathcal{SH}_{\text{best}}$ ▷ the best model so far
  $E_{\text{best}} \leftarrow \infty$ ▷ the best error so far
  $\mathcal{I}_{\text{best}} \leftarrow \emptyset$ ▷ the best set of inliers so far
  while $i < k$ **do**
    draw a random minimal set $\mathcal{S}$ of n samples from $\mathcal{P}$
    instantiate a new parametrized model $\mathcal{SH}(\boldsymbol{\beta})$ from $\mathcal{S}$
      ▷ generate a model hypothesis
    compute the error $E(\mathcal{SH})$ and the inlier set $\mathcal{I}(\mathcal{SH})$
      of the new model $\mathcal{SH}$ from $\mathcal{P}$
    if $E(\mathcal{SH}) < E_{\text{best}}$ **then** ▷ whether the hypothesis is better
      $\mathcal{SH}_{\text{best}} \leftarrow \mathcal{SH}$
      $E_{\text{best}} \leftarrow E(\mathcal{SH})$
      $\mathcal{I}_{\text{best}} \leftarrow \mathcal{I}(\mathcal{SH})$
    end if
    update k with $|\mathcal{I}(\mathcal{SH})|$
    $i \leftarrow i + 1$
  end while
  return $(\mathcal{SH}_{\text{best}}, E_{\text{best}}, \mathcal{I}_{\text{best}})$
end function

This optimization scheme is very flexible and not only LMS, RANSAC and their variants but also the RHT can be described within this framework. In RANSAC, a minimum number of inliers is often specified so that $E(\mathcal{SH}) = \infty$ in Algorithm 2 if the model hypothesis $\mathcal{SH}(\boldsymbol{\beta})$ does not have enough inliers. As a result, no acceptable model may be found after all hypotheses have been generated. The optimization may also not find the optimal model, either because not enough hypotheses have been generated or because the instantiated models do not explain well enough the set of measurements. The outlined random sampling scheme can be modified in a number of ways:

- the drawing of samples does not need to be uniform and may be altered or even biased to favor some model hypotheses,
- the instantiation of models from minimal sets can also be altered,
- the error $E(\mathcal{SH})$ (the objective function) can be modified (in particular

to evaluate to ∞ in order to invalidate some hypotheses),
- optionally, the maximum number of iterations can be modified (typically decreased) by taking into account each newly discovered set of inliers $I(\mathcal{Sh})$.

Some parameters for random sampling in general or RANSAC in particular need to be specified and are now detailed.

Maximum number of iterations Model hypotheses can not be generated indefinitely and the number of iterations has to be limited. The number of samples to draw can be derivated in different ways. First, assuming an inlier ratio of w and uniformly random drawings, the expected value $\mathrm{E}[k]$ of the minimum number of iterations k is[1] $\mathrm{E}[k] = w^{-n}$ and its standard deviation is $\sigma[k] = \frac{\sqrt{1-w^n}}{w^n}$. The minimum number of iterations may then be chosen as $\mathrm{E}[k] + \lambda\,\sigma[k]$ where λ weights the confidence in the chosen number of iterations. A more standard derivation estimates the probability p of success after k drawings of minimal sets: $p = 1 - (1 - w^n)^k$, which leads to:

$$k = \frac{\log(1-p)}{\log(1-w^n)} \; .$$

As noted before, the assumed inlier ratio w can be updated each time a set of inliers $\mathcal{I}(\mathcal{Sh})$ is found and k be consequently re-estimated. Note that the number of iterations only depends on the ratio of inliers and not on the total number of points.

Minimum size of the consensus set (RANSAC) A common option is to chose a minimum size close to the assumed number of inliers wN, but w is often only a guess and this size threshold may be problem-specific.

Distance threshold (RANSAC) For lack of a better error model, if the measurement error relative to a model is assumed to follow a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with zero mean and σ standard deviation, a distance threshold $d_{\max}$ to decide whether a point is an inlier or not can be estimated with a given confidence. The square of the point to model distance $r_i^2(\boldsymbol{\beta}) = d^2(P_i, \mathcal{Sh}(\boldsymbol{\beta}))$ is a squared Gaussian variable and thus follows a χ_1^2 distribution. The probability that the value of a χ_1^2 random variable is less

[1]

$$\begin{aligned} \mathrm{E}[k] &= \sum_{k=1}^{\infty} k\,(1-w^n)^{k-1}\,w^n \;=\; \ldots \;=\; w^{-n}\,, \\ \mathrm{Var}[k] &= \sum_{k=1}^{\infty} \left(k - w^{-n}\right)^2 (1-w^n)^{k-1}\,w^n \;=\; \ldots \;=\; \frac{1-w^n}{w^{2n}}\,. \end{aligned}$$

than x^2 is given by the cumulative χ^2_1 distribution $F_1(x^2) = \int_0^{x^2} \chi^2_1(y)\,\mathrm{d}y$. For the probability 0.95 that a point is an inlier, the distance threshold to use is then $d^2_{\max} = F_1^{-1}(0.95)\,\sigma^2 \approx 3.84\,\sigma^2$. For a probability of 0.99, the threshold becomes $d^2_{\max} \approx 6.63\,\sigma^2$.

Random sampling methods and multiple structures

As noted above, the estimation of models with RANSAC is biased when data sets feature multiple structures acting as structured outliers or pseudo-outliers. Extending random sampling methods for multiple structures regression is challenging. Some attempts include the incorporation of robust statistical measures [Chen et al., 2001] to estimate noise scale. The multi-RANSAC method of [Zuliani et al., 2005] adapts RANSAC to simultaneously recover several models but the number of models has to be specified. Other recent approaches [Zhang and Kosecká, 2006, Toldo and Fusiello, 2008] try exploiting the clustering of hypotheses in the same spirit as the detection of modes in parameter space but use instead the distribution of residuals or consensuses.

We do not pretend to solve the general problem of multiple structures segmentation from noisy point clouds with outliers. In the next section, we propose to exploit additional priors from our problem (the acquisition process, the geometry of the extracted shapes and the assumed higher density of the point cloud near the shapes) to significantly improve the robustness of existing approaches.

6.2 Single shape robust extraction

Our shape detection and fitting method is based on the RANSAC framework of [Fischler and Bolles, 1981]: random shapes instantiated by minimal sets of points[2] are enumerated to optimize an objective function counting the number of points (the shape inliers) inside a band around a shape instance. As reminded in the previous section, plain RANSAC is not suited to robust regression for data with multiple structures. The commonly used approach to multiple model detection is to extract a model with RANSAC, remove the inliers to the extracted model and iterate. Iterating can lead to catastrophic results. Figure 6.4 shows the result of a sequence of successive shape extractions with the MSAC [Torr and Zisserman, 2000] variant of RANSAC: after one or two successful extractions, MSAC begins to hallucinate structures that slice the whole point cloud with a good fit. In contrast, our approach does not suffer from this problem. Even if it is derivated from RANSAC and if simple successive shape detections are applied, our approach successfully recover all the shapes with the same number of model hypotheses.

[2] 3 points for a plane, 4 points for a sphere...

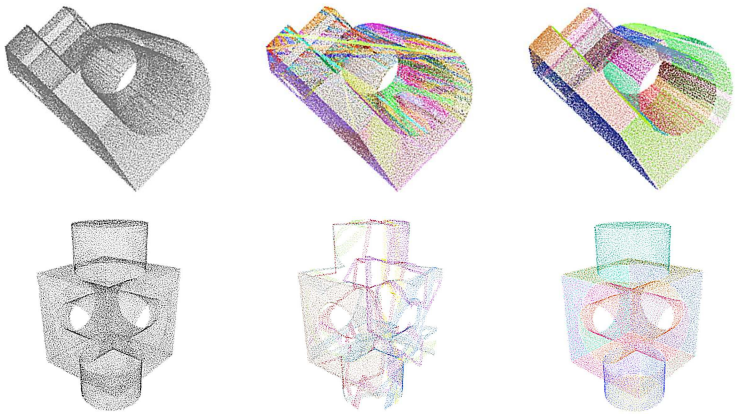


Figure 6.4: ***Sequential MSAC:*** *the sequential extraction of shapes with naive RANSAC methods may lead to catastrophic cascaded errors (both *SAC methods used the same number of iterations). First row: extraction of planes only on the* joint *data set. From left to right: noisy input point cloud, sequential MSAC extraction of planes, our improved MSAC extraction. Second row: extraction of different shapes on the* block *data set. From left to right: noisy input point cloud, sequential MSAC extraction of shapes, our improved MSAC extraction.*

Before any shape detection occurs, a 3D k-D tree [Bentley, 1975] of the point cloud is first computed to efficiently find the k nearest neighbors (abbreviated k-NN, $k = 10$ in practice) and estimate the normal of each point by fitting a plane to these k-NN with a least-squares estimate. The normal orientation $\overrightarrow{n}_P$ is corrected using the visibility information v_P of the sample point P. If the surface has been densely enough sampled by the point cloud $\mathcal{P}$, the k-NN of a point near the surface are likely to also lie near the surface, leading to a more reliable normal estimation. The k-NN of an outlier point, however, are much more spatially spread and its normal will likely be incoherent with its neighbors'. This information will be put to use to filter out outliers.

As noted in the description of random sampling optimization, the random sampling scheme is quite flexible and can be changed in a number of ways.

Objective function The objective function or error to be minimized over all the hypothesized models from one class is modified as follows:

1. being based on the MSAC variant [Torr and Zisserman, 2000] of RANSAC, it penalizes inliers according to their distances to the instantiated surface.
2. The acquisition process intervenes in the inlier counting procedure: a point is considered an inlier only if its visibility information agrees with the instantiated shape, *i.e.* , if the local oriented normal to this shape

does not make a wide angle with any of the lines of sight of the point.
3. The local surface oriented normal is similarly considered in the inliers counting procedure: a point is an inlier only if its oriented normal is close to the local normal of the instantiated shape.
4. Finally, the inlier counting procedure exploits the graph induced by the k-NN relation to count the number of inliers in the largest connected regions from the seed points (that instantiated the surface) and inside the inliers band. Such region growing approach helps avoiding the so-called "hallucination" problem of RANSAC variants (artificially finding structure by relating small clusters) by naturally handling densely sampled multiple structures while keeping the robustness advantages of a RANSAC-based optimization. It is also more efficient since only a subset of the points are visited.

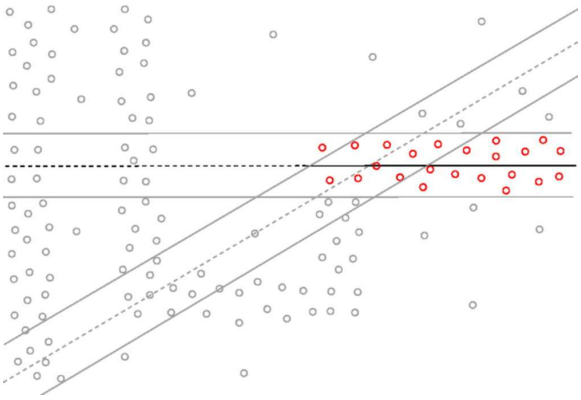


Figure 6.5: ***The modified RANSAC objective function*** *exploits a supposed spatial proximity of inliers and the asymmetric k-NN relation to avoid the so-called "hallucination" of structures (illustrated by the diagonal band).*

To summarize these modifications, a point P is an inlier (and the predicate $\textsc{Inlier}_{\mathcal{SH}}[P]$ is true) if and only if its normal $\overrightarrow{n}_P$ is, up to some tolerance (typically $\theta_{\max} = 20°$), aligned with the local normal to the shape $\mathcal{SH}$, if its line-of-sights are all correctly oriented *w.r.t.* the normal of the shape $\overrightarrow{n}_{\mathcal{SH}}(P)$ at P (with a tolerance $\gamma_{\max} = 85°$ in practice) and if it lies in a band define by $d_{\max}$ around the instantiated shape $\mathcal{SH}$:

$$\begin{aligned}\textsc{Inlier}_{\mathcal{SH}}[P] \quad \Leftrightarrow \quad & \big(d(\mathcal{SH}, P) \leq d_{\max}\big) \\ & \wedge \big(\overrightarrow{n}_P . \overrightarrow{n}_{\mathcal{SH}}(P) \geq \cos\theta_{\max}\big) \\ & \wedge \big(\forall Q \in v_P \quad \overrightarrow{PQ} . \overrightarrow{n}_{\mathcal{SH}}(P) \geq \cos\gamma_{\max} \|PQ\|\big) \ .\end{aligned}$$

Now, we note $P \longrightarrow Q$ the relation "the point Q is a k-NN of the point P" and $P \xrightarrow{\mathcal{SH}} Q$ the relation $(P \longrightarrow Q) \wedge \textsc{Inlier}_{\mathcal{SH}}[Q]$. The transitive closure of the relation $\xrightarrow{\mathcal{SH}}$ on $\mathcal{P}$ from a subset $\mathcal{U} \subseteq \mathcal{P}$ of points is noted $\xrightarrow{\mathcal{SH}}{}^{\star}$ and defined as:

$$\mathcal{U} \xrightarrow{\mathcal{SH}}{}^{\star} Q \quad \Leftrightarrow \quad \left\{ \begin{array}{ll} \exists P \in \mathcal{U} & P \xrightarrow{\mathcal{SH}} Q_1 \xrightarrow{\mathcal{SH}} \dots \\ \exists p \in \mathbb{N} \quad \exists Q_1, \dots, Q_p \in \mathcal{P} & \dots \xrightarrow{\mathcal{SH}} Q_p \xrightarrow{\mathcal{SH}} Q \end{array} \right. ,$$

i.e. , the point Q can be "reached" through a sequence of k nearest neighbors that are also inliers from a point $P \in \mathcal{P}$. The new objective function we consider is formally:

$$\boldsymbol{\beta}^{\star} = \underset{\substack{\mathcal{U}=\{P_{i_1},\dots,P_{i_m}\}\subseteq\mathcal{P} \\ \mathcal{U}\subseteq\mathcal{SH}(\boldsymbol{\beta})}}{\arg\min} \sum_{i=1}^{N} \rho\left(\frac{r_i(\boldsymbol{\beta})}{r_{\max}}, P_i, \mathcal{U}\right) ,$$

with $\rho(x, Q, \mathcal{U}) = \left\{ \begin{array}{ll} x^2 & \mathcal{U} \xrightarrow{\mathcal{SH}}{}^{\star} Q \\ 1 & \text{else} \end{array} \right.$. m is the minimal number of points $|\mathcal{U}|$ to instantiate a parametrized shape $\mathcal{SH}(\boldsymbol{\beta})$. While this objective function has some similarity with previous ones, it explicitly accounts for the optimization method (random sampling of minimal sets), and it asks for the local proximity of inliers as shown in Figure 6.5. The loss function depends on both the point and the minimal set and not only on the residual of the point. It could also depend on the normal and line-of-sight used to decide whether a point is an inlier, *à la* MSAC. We actually experimented with it but found no significant difference in practice.

Finally, we verify that, as announced, this objective function is efficiently computed with a simple (breadth-first or depth-first, for instance) search algorithm from seed points in the oriented k-NN induced graph by checking whether each vertex is an inlier and limiting only this search to inliers. This effectively amounts to finding the largest connected regions of inliers from the seed points $\mathcal{U}$.

Shape hypotheses The way samples are drawn and shapes are instantiated is also modified to consider more meaningful hypotheses.

- The density of samples is supposed to be higher near the surface. The closer the points, the higher the chance the points are inliers to the same shape. Of the few points to be randomly selected to create a shape instance, the first point P_{i_1} is drawn uniformly in the point cloud, while the next are uniformly drawn but only within $\mathcal{B}(P_{i_1}, r)$, a ball of small radius $r = s\, d_{\max}$ (a fixed multiple of the maximum inlier distance $d_{\max}$). This geometric ball search can be answered as a

range query again in the k-D tree of the points. While this operation makes the plane sampling a more involved operation, this cost is outweighed by improving the chance of drawing meaningful shapes and allowing the maximum number of iterations of the random sampling to be significantly reduced. This localized sampling follows NAPSAC [Myatt et al., 2002], developed in the context of high-dimensional robust estimation, where an assumed distribution of inliers and outliers led to this idea. In theory, NAPSAC would require sorting the samples by increasing order of distance to P_{i_1} to pick additional samples with an adequate distribution. In practice, the authors also only uniformly sample points inside a hypersphere of fixed radius derived from the inferred distributions of inliers and outliers.

- Only shape hypotheses whose minimal sets of points agree on both visibility and local normal orientation are drawn. The points $P_{i_1}, \ldots, P_{i_m}$ from a minimal set $\mathcal{U}$ have to be inliers to the instantiated $\mathcal{SH}(\boldsymbol{\beta})$ according to the inlier definition given above: $\forall i \in [\![i_1, i_m]\!] \quad \text{INLIER}_{\mathcal{SH}(\boldsymbol{\beta})}[P_i]$.
- The point clouds from Chapter 1 may still be quite noisy (especially in the passive stereo case) and random shapes exactly supported by minimal sets of point lead to systematic wrong hypotheses (with a possibly wrong optimal shape as a consequence), especially for non-planar shapes. An improved estimate of the instantiated random shape $\mathcal{SH}$ is obtained by refitting it to all the local inlier points within the band around $\mathcal{SH}$ (restricted to the ball $B(P_1, r)$ used to search the minimal set) with distance loss weighted least-squares.

Note that this shape extraction method is general and hence applicable to any class of orientable surface for which approximate distance and normal can be computed. Also note that all the involved geometric queries (ball query and k nearest neighbors) rely only on the initial computation from the k-D tree from the input point cloud.

Shapes of different classes (only planes in Chapter 7 and planes, spheres, cylinders and cones in Chapter 8) are tentatively extracted from the point set, and the shape with the lowest error is selected (provided there is one). The particular choice of distance to the shapes has been omitted in this discussion. While computationally attractive, fitting using the simple algebraic signed distance $f_{\boldsymbol{\beta}}$ for implicit surfaces ($f_{\boldsymbol{\beta}}(P) = 0$ if $P \in \mathcal{SH}(\boldsymbol{\beta})$) is known to be biased (in the presence of noise and in particular when the shape is partially occluded). Instead several first and higher order approximations have been proposed [Taubin, 1988, 1991, 1993] as a compromise between speed and accuracy. We depart from this approach and use the Euclidean distance to a shape. In the cases considered in the next two chapters, this distance has a closed form. Except for planes, the fitting optimization is a non-linear least-squares problem that we solve through an iterative optimization. The particular details of shape distance, instantiation and fitting for each class of shapes are described in Appendix A. Finally, even if all second order shapes

used in Chapter 8 could be reduced to a single, more generic class of quadrics, we explicitly distinguish the shapes, to reduce the number of parameters and the computational cost of generating shape hypotheses, to allow the computation of Euclidean distance in closed form (for general quadrics this would require finding a root of a 6-th order polynomial) as well as to avoid having to handle geometric degeneracies.

6.3 Hierarchical extraction of multiple shapes

Instead of repeatedly applying the above single shape extraction by sequentially removing fitted point from the point set, the extraction is modified so as to guide the detection towards interesting shapes and simultaneously build the BSP tree required for the surface reconstruction step. At first the localized sampling described above helping locating large dominant shapes in the whole point cloud. As shapes are recovered the corresponding spatial subdivision then progressively assumes the role of the localized sampling to extract meaningful shapes. A summary of the whole method is shown in Algorithm 6.3 where ExtractShape corresponds to the single shape extraction described above.

Initially, the BSP tree is reduced to a single leaf for the whole space. From a previously built BSP tree, a restricted shape extraction is tried in turn in each active leaf. If no shape can be extracted from a leaf, the leaf is marked as inactive and is not be explored anymore. After a successful extraction, a few steps are followed. Sufficiently large connected sets of inliers (as in Section 6.2) are found in the band around the detected shape. With least-squares weighted by shape distance loss, the surface is refitted to all these new inliers points which are now excluded from the point cloud. Outliers lying in the band are however kept for further shape sampling (distinguishing points used for sampling and error is done to prevent overfitting while still allowing interesting shape hypothesis to be generated). Finally all the leaves of the tree intersected by the shape clusters are split and new nodes corresponding to the shape are created along with two leaves. The whole process is iterated until no further shape extraction is possible. If needed, the location of the points (the leaf they belong to) is updated at each leaf splitting event.

The described search and split strategy may also be modified in a number of ways:

- If all the leaves intersected by the recovered shape are systematically split, the resulting spatial subdivision is the so-called arrangement of the shapes and contains many superfluous regions that are not relevant for the final reconstruction and would make the application of the surface fitting framework of Chapter 2 more difficult. Splitting only leaves with enough shape supported points is preferable both to make the final surface reconstruction better posed and to reduce its

computational cost.

- At the other extreme, splitting only the leaf where the search has occurred may avoid possible misclassification or poorer fit but the possibility of identifying large shared unconnected components of the same shape is then lost.
- Finally, if the shape hypotheses are always formulated from points of the whole point cloud, a larger number of hypothesis may be required in presence of outliers. Restricting the search to leaves that are invalidated when no shape can be extracted is more efficient and further lowers the required number of iterations.

Experiments demonstrating the hierarchical shape extraction on synthetic and real data are postponed to Chapters 7 and 8 so as to show the whole segmentation and reconstruction procedure from point clouds at once.

6.4 Conclusion

This chapter has introduced the first step in our approach to the problem of surface reconstruction with strong shape priors from noisy points with outliers.

We proposed several improvements to the classical random sampling framework applied to shape fitting. These improvements allow the robust extracting of a shape in a dense point cloud even in the presence of multiple structures. The key to this achievement is to take into account the acquisition process, the local geometry of the shape and the spatial proximity of inliers both in the sampling process and in the error associated to a shape instance.

The single shape extraction is then included in a hierarchical shape decomposition algorithm that given a point cloud outputs a BSP tree-like description of a subset of the points into multiple shapes. This hierarchical decomposition is used in the next two chapters for the final surface reconstruction.

It should be noted that this whole shape fitting procedure is generic and can be extended to several classes of orientable surfaces for which approximate distance and normal can be computed.

Algorithm 3 Synopsis of the shapes extraction and tree construction algorithm

function ExtractShapes($\mathcal{P}$, ...)
  compute the k nearest neighbors of each point
  estimate the normal to each point from its neighbors
  $\mathcal{T} \leftarrow$ a single leaf ▷ the BSP tree
  $\mathcal{SH} \leftarrow \emptyset$ ▷ the set of extracted shapes
  $\mathcal{P}_{\text{samp}} \leftarrow \mathcal{P}$ ▷ mark all points as used for sampling
  $\mathcal{P}_{\text{err}} \leftarrow \mathcal{P}$ ▷ mark all points as used for error
  mark the single leaf of $\mathcal{T}$ as active
  while there is an active leaf in $\mathcal{T}$ **do**
    for all active leaves of $\mathcal{T}$ **do**
      $\ell \leftarrow$ the current active leaf of $\mathcal{T}$
      $\mathcal{P}^{\ell}_{\text{samp}} \leftarrow \mathcal{P}_{\text{samp}} \cap \ell$
      $\mathcal{P}^{\ell}_{\text{err}} \leftarrow \mathcal{P}_{\text{err}} \cap \ell$
        ▷ points used for sampling and error restricted to ℓ
      $\left(\mathcal{Sh}^{\ell}, E^{\ell}, \mathcal{I}^{\ell}\right) \leftarrow$ ExtractShape($\mathcal{P}^{\ell}_{\text{samp}}$, $\mathcal{P}^{\ell}_{\text{err}}$)
        ▷ (tentatively) extract a new shape $\mathcal{Sh}$ in ℓ
      if $E^{\ell} = \infty$ **then**
        mark the leaf ℓ as inactive
        continue to the next active leaf in $\mathcal{T}$
      end if
      $\mathcal{I} \leftarrow$ the (large) connected sets of inliers to $\mathcal{Sh}^{\ell}$ in $\mathcal{P}_{\text{samp}}$
      $\mathcal{O} \leftarrow$ outliers in $\mathcal{P}_{\text{samp}}$ lying in the band around $\mathcal{Sh}^{\ell}$
      estimate $\mathcal{Sh}$ from $\mathcal{I}$ and its initial estimate $\mathcal{Sh}^{\ell}$
      $\mathcal{SH} \leftarrow \mathcal{SH} \cup \{\mathcal{Sh}\}$
      mark all the points of $\mathcal{I}$ as belonging to $\mathcal{Sh}$
      $\mathcal{P}_{\text{samp}} \leftarrow \mathcal{P}_{\text{samp}} \setminus \mathcal{I}$
      $\mathcal{P}_{\text{err}} \leftarrow \mathcal{P}_{\text{err}} \setminus (\mathcal{I} \cup \mathcal{O})$
        ▷ note $\mathcal{P}_{\text{err}} \subseteq \mathcal{P}_{\text{samp}}$ is thus always verified
      for all leaves of $\mathcal{T}$ intersected by $\mathcal{I}$ **do**
        $\ell' \leftarrow$ the current leaf of $\mathcal{T}$ intersected by $\mathcal{I}$
        turn the leaf ℓ' into a internal node of $\mathcal{T}$ associated to $\mathcal{Sh}$
        add two new leaves ℓ'_1 and ℓ'_2 to this internal node
        mark the leaves ℓ'_1 and ℓ'_2 as active
      end for
    end for
  end while
  return $(\mathcal{T}, \mathcal{SH})$
end function

CHAPTER 7

Concise piecewise-planar surface reconstruction

This chapter applies the shape extraction method of the previous chapter to classes of shapes restricted to planes only. In this case applying a variant of the surface fitting framework of Chapter 2 to a polyhedral complex induced by the hierarchical description of the point cloud leads not only to geometrically simple models but also to very concise ones with extremely low combinatorial complexity. Promising results of the combined shape extraction and surface reconstruction approach are shown both on synthetic and real data.

Contents

Our first application of the hierarchical shape extraction procedure is restricted to planes and produces a piecewise-planar hierarchical description of the point set $\mathcal{P}$. The surface reconstruction uses as input, a subset $\mathcal{P}' \subseteq \mathcal{P}'$ of the original input cloud along with the cluster representative $\mathcal{Pl}_P$ (the plane) to which each point P belongs and the tree description.

7.1 Surface reconstruction algorithm

The hierarchical segmentation induces a particular cell complex and with the notations introduced in Chapter 2, a pseudo-surface can be extracted corresponding to a globally optimum labeling of cells *w.r.t.* the following energy:

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}'}(\mathcal{S}, \mathcal{P}', v) \ ,$$

where $\mathrm{E}_{\mathrm{vis}'}$ is a variant of $\mathrm{E}_{\mathrm{vis}}$ that penalizes surfaces crossing a line of sight and surface mis-orientations *w.r.t.* to the clustered points $\mathcal{P}'$. Optionally, for difficult point clouds from stereo, an additional photo-consistency term is used to favor photo-consistent facets of the complex and to compensate for poorly sampled geometry (caused by the lack of texture) or to avoid structured outliers:

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}'}(\mathcal{S}, \mathcal{P}', v) + \lambda_{\mathrm{photo}} \, \mathrm{E}_{\mathrm{photo}}(\mathcal{S}) \ ,$$

as in Chapter 4, $\mathrm{E}_{\mathrm{photo}}(\mathcal{S})$ is the sum of the photo-consistencies of the oriented facets composing $\mathcal{S}$ (using for instance the fitted points close to the considered facet to determine which views to use):

$$\mathrm{E}_{\mathrm{photo}}(\mathcal{S}) = \sum_{\substack{C_i, C_j \in \mathcal{C}_3 \\ f^\star_{C_i}=0, f^\star_{C_j}=1 \\ C_i \cap C_j = F}} 1 - \rho\left(F^{C_j \to C_i}\right) \ .$$

In the presented results for multi-view stereo, this computationally expensive term was not necessary and was not used.

7.1.1 BSP polyhedral complex

The choice of cell complex is imposed by the hierarchical segmentation of the input point clouds in structures. Since the detected shapes are restricted to planes, the hierarchical structure is a standard BSP tree [Fuchs et al., 1980]. The corresponding cell complex is a polyhedral complex that can be computed exactly with a simple modification of an algorithm for the incremental construction of hyperplane arrangements: the associated details are given in Appendix B. Examples of such cell complexes are shown in Figure 7.1 for the extractions of planar structures on the data sets of Figures 7.3

and 7.10. The extracted pseudo-surface is a mesh with or without boundaries, with convex polygonal facets and identified supporting (and possibly shared) planes. Note that while the results shown in Section 7.2 are simple, the optimization domain is not trivial and only embeds the recovered surface: the selected facets often represent much less than half the facets of the embedding complex (see Tables 7.1 and 7.2).

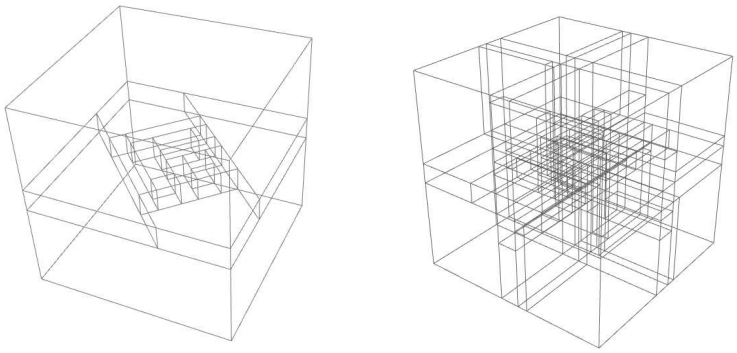


Figure 7.1: ***BSP polyhedral complexes:*** *the final surface is extracted as a union of facets in a polyhedral complex. Left: polyhedral complex for the* double torus II *data set, right: for the* cubes *data set.*

Very recently, we have discovered the work of [Murali and Funkhouser, 1997] where an inside/outside labeling of the cells of a (plane based) BSP complex is used to recover a simple polygonal model in the quite different context of remeshing noise-free polygon soups. In this work the labeling is based on the facet coverage of polygons to define a notion of cells "solidities" and reduces to the inversion of a sparse matrix. The proposed formulation is not suited to data sets with noise and outliers and cannot recover open scenes.

7.1.2 Surface visibility

The visibility term E_{vis} of Chapter 2 is slightly adapted to sets of points which are known to belong to clusters of points from an extracted plane. These points are located near a facet of the complex whose supporting plane corresponds to their plane cluster as facets of the complex are built from plane clusters. The considered sequence of traversed cells by a line of sight is thus modified to account for this additional knowledge: the sequence is stopped at the facet $F \in \mathcal{C}_2$ of the complex corresponding to the plane $\mathcal{Pl}_P$ of the fitted point P, provided this facet exists. Using the same notations as in Chapter 2, the visibility constraints are written as:

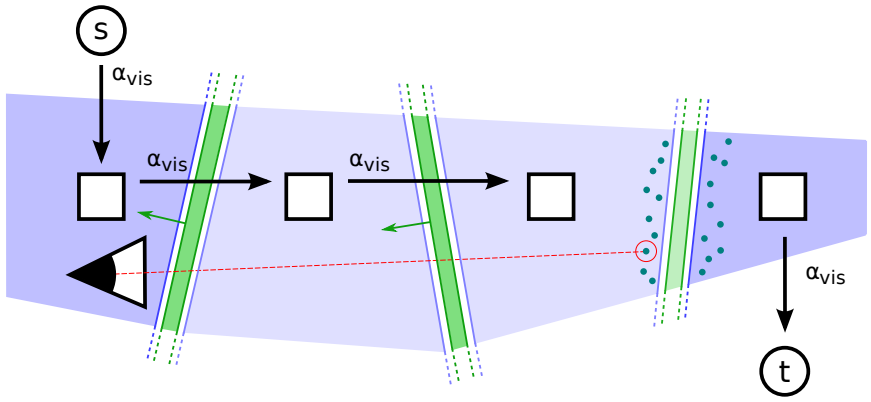


Figure 7.2: ***Visibility.*** *A fitted point (red circled dot) affects the oriented facets (darker green) and cells (darker blue) weights along one of its lines of sight (red dotted line) from its plane cluster.*

$$\mathrm{E}_{\mathrm{vis}'}(\mathcal{S},\mathcal{P}',v) = \sum_{P\in\mathcal{P}'} \sum_{\substack{Q\in v_P \\ \exists F\in C_2 \\ F\subset\mathcal{P}_P \wedge \dim([PQ]\cap F)=0}} D_{\mathrm{out}}\left(l_{C_1^{Q\to P}}\right) + \sum_{i=1}^{N_{[QP]}-1} V_{\mathrm{align}}\left(l_{C_i^{Q\to P}}, l_{C_{i+1}^{Q\to P}}\right) + D_{\mathrm{in}}\left(l_{C_{N_{[QP]}+1}^{Q\to P}}\right) .$$

This term is obviously suitable to minimum s-t cut optimization. As shown in Figure 7.2, the oriented facets crossed by the considered line of sight (darker green) get a weight of α_{vis}, while the cell (darker blue) where the camera optical center lies is linked to the source s with an α_{vis} weight and the cell behind the plane of the fitted point (darker blue) is linked to the sink t with a weight α_{vis}. For all the available lines of sight, these weights or penalties for cells being inside or outside the surface and facets being part of the reconstruction are accumulated.

7.2 Experimental results

The running time of the prototype implementation seems reasonable for the presented results: it varies from 1m to 15m for the extraction of the planes (conservative settings were used) and the surface reconstruction (including

the exact computation of the polyhedral complex) typically takes less than 10m and is dominated by the ray traversal of the complex.

Noisy point clouds (synthetic) Several experiments were run on synthetic data to evaluate both the segmentation step of Chapter 6 and the reconstruction step of this chapter. Each time a point cloud is generated from the vertices of an original mesh. The visibility of the points *w.r.t.* 64 sources located around the object is evaluated with this ground truth model to check for occlusions. The point location are contaminated with isotropic Gaussian noise. Figures 7.3, 7.5, 7.7 and 7.8 show segmentations and reconstructions on a variety of different point clouds. Curved geometry is approximated with several planar patches Figure 7.9 depicts how different settings of the tolerance parameters of the plane detection lead to various levels of details for the reconstructed surface. In Figure 7.6, the concise reconstruction of the *double torus I* and *double torus II* data sets is compared both in quality and conciseness with Poisson surface reconstruction [Kazhdan et al., 2006] and the reconstruction method of Chapter 4. Not only our output mesh is almost unaffected by the added noise, but it is several order of magnitude less complex than both other reconstructions and recovers the sharp features of the original model.

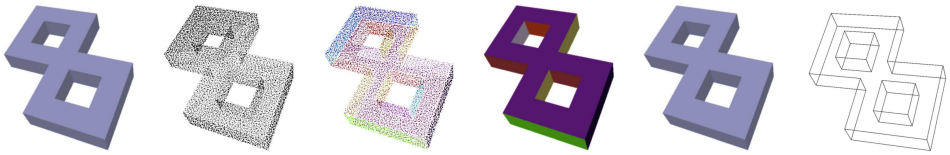
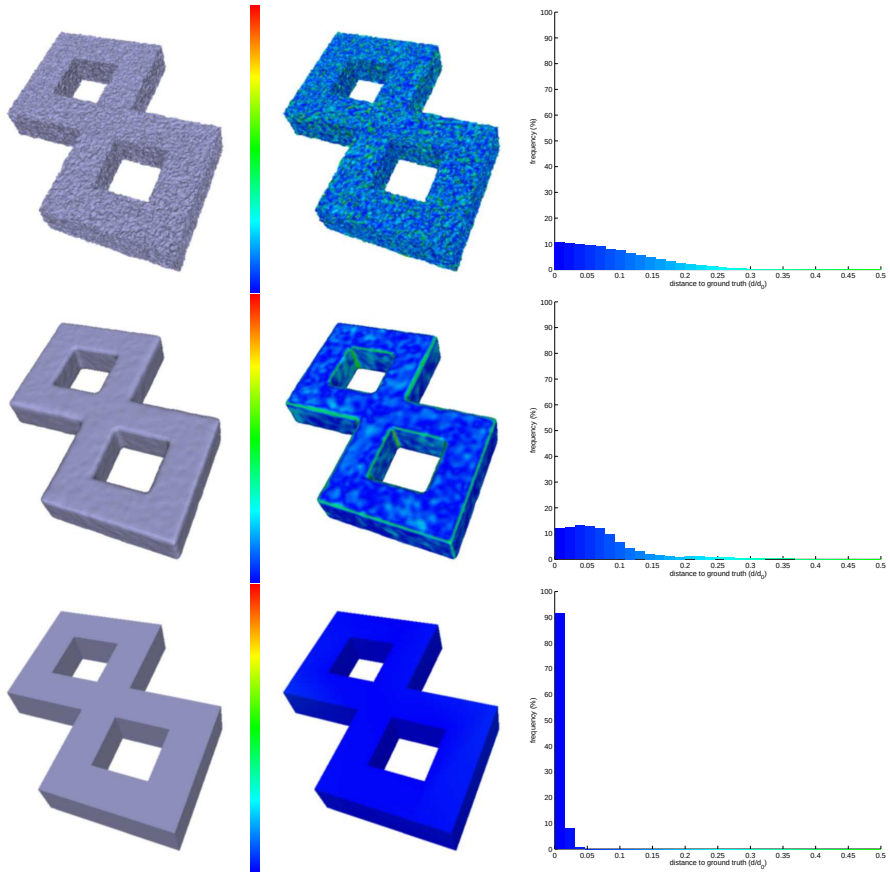


Figure 7.3: ***double torus I:*** *our reconstruction is visually indistinguishable from the ground truth. From left to right: ground truth mesh, noisy input point cloud, segmented point cloud and concise piecewise-planar reconstruction (colored by plane / uncolored / wireframe contours).*

***cubes* (multi-view stereo)** The experiment of Figure 7.10 uses 60 rendered images (1 Mpix) of 6 cubes and the input point cloud is generated with the method described in Chapter 1 and based on interest points. The matching ambiguity of the texture (the same on all the cubes) results in a very noisy point cloud with a fair amount of outliers (only half of the input point cloud was clustered into planes by our method). Despite this, as shown in Figure 7.11, our algorithm outputs a surface that has the expected complexity of the original model. In contrast, Poisson surface reconstruction fails to output a satisfying reconstruction due to the insufficient coherence of the estimated normals. The reconstruction method of Chapter 4 again out-



		#Points	#Vertices	#Facets
	Original mesh		39K	78K
	Input point cloud	39K		
Reconstructions:	Delaunay, $E(\mathcal{S}) = E_{vis}(\mathcal{S}) + \lambda_{qual} E_{qual}(\mathcal{S})$		36K	72K
	Poisson [Kazhdan et al., 2006]		21K	42K
	Concise reconstruction, $E(\mathcal{S}) = E_{vis'}(\mathcal{S})$		46	41

Figure 7.4: ***double torus I data set:*** *comparison with the general purpose reconstruction method of Chapter 4 and Poisson surface reconstruction. First row: Delaunay, second row: Poisson, third row: concise reconstruction. First column: reconstructed surface, second column: reconstructed surface (colored by distance to ground truth), third column: histogram of distances to ground truth.*

puts a noisy and much more complex mesh. The high level of noise however makes our method poorly estimates one of the 18 extracted planes.

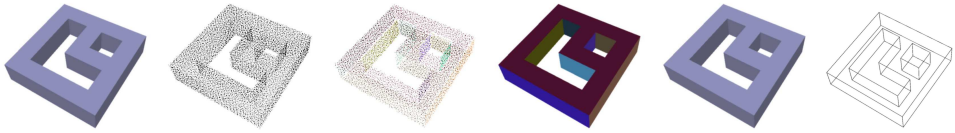


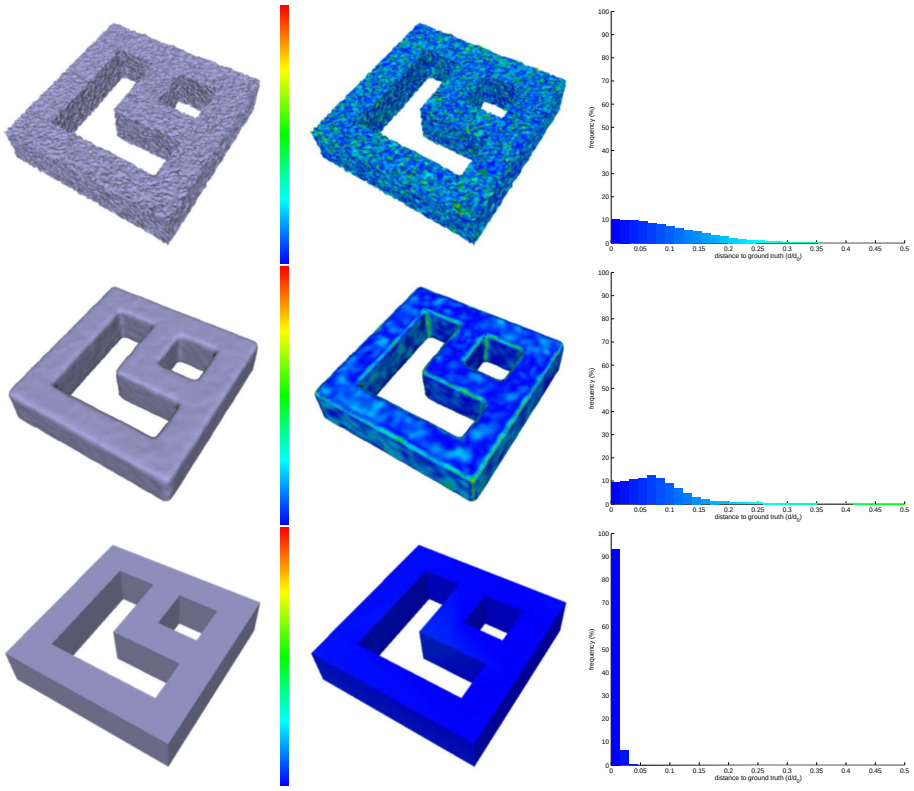
Figure 7.5: ***double torus II:*** *our reconstruction is visually indistinguishable from the ground truth. From left to right: ground truth mesh, noisy input point cloud, segmented point cloud and concise piecewise-planar reconstruction (colored by plane / wireframe contours).*

		double torus I	*double torus II*	*block*	*fandisk*	*joint 1*	*joint 2*	*joint 3*
Original mesh	#Vertices	39K	26K	49K	44K		102K	
	#Facets	78K	51K	99K	89K		204K	
Input point cloud	#Points	39K	26K	49K	44K		102K	
	σ_{noise}	0.05	0.05	0.07	0.01		0.0025	
	ℓ_{max}	20	20	38	5.25		1.12	
Segmentation	#Extracted planes	14	16	34	53	35	54	71
	#Clustered points	34K	17K	45K	38K	94K	91K	90K
		86%	68%	92%	86%	92%	89%	88%
Reconstruction	#Vertices (complex)	82	72	317	462	189	268	342
	#Facets (complex)	107	86	435	646	247	348	449
	#Vertices (surface)	46	48	222	280	139	203	260
		56%	67%	70%	61%	74%	76%	76%
	#Facets (surface)	41	40	178	218	112	155	198
		50%	47%	41%	34%	45%	45%	44%

Table 7.1: ***Detailed input and results:*** *data for the results on the synthetic data sets.* σ_{noise} *is the standard deviation of the added isotropic Gaussian noise and* ℓ_{max} *is the length along the maximum dimension of the bounding box of the point cloud.*

***rue Soufflot* (range data)** In Figure 7.12, our method is tested on range data acquired from a mobile car equipped with a GPS/IMU unit paired with a time-of-flight range driving in a street along a block (a part of the data set featured in Chapter 3 without occlusion). This demonstrates a potential application of our approach to reconstruct simple models of whole streets from ground acquired data even in the presence of numerous non-planar features (pedestrians, terraces, ...). The method is however limited, and not applicable at once to reconstruct the whole street, since adjacent streets are half-occluded and the corresponding planes thus cannot be extracted.

***ALOI #418* (multi-view stereo)** The Amsterdam library of object images [Geusebroek et al., 2005] offers a large collection of small objects cap-



		#Points	#Vertices	#Facets
	Original mesh		26K	51K
	Input point cloud	26K		
Reconstructions:	Delaunay, $\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}}(\mathcal{S}) + \lambda_{\mathrm{qual}}\,\mathrm{E}_{\mathrm{qual}}(\mathcal{S})$		24K	49K
	Poisson [Kazhdan et al., 2006]		22K	43K
	Concise reconstruction, $\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}'}(\mathcal{S})$		48	40

Figure 7.6: ***double torus II data set:*** *comparison with the general purpose reconstruction method of Chapter 3 and Poisson surface reconstruction. First row: Delaunay, second row: Poisson, third row: concise reconstruction. First column: reconstructed surface, second column: reconstructed surface (colored by distance to ground truth), third column: histogram of distances to ground truth.*

tured under varying viewing angle, illumination angle and illumination color. Of particular interest here, is the object #418, an ornamental stone and an almost purely piecewise-planar object. The 72 images (0.5 Mpix) of this object undergoing a rotation were calibrated using a tracking software. A point cloud from matched interest points was then computed with the method of Chapter 1. The output of our segmentation and reconstruction algorithm is

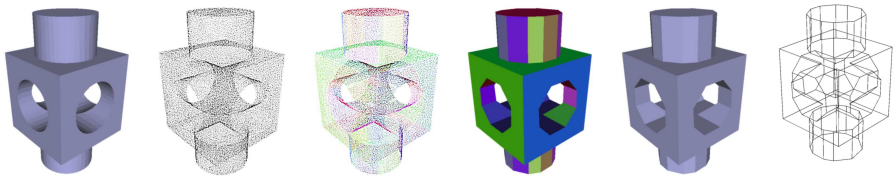


Figure 7.7: ***block data set:*** *ground truth mesh, noisy input point cloud, segmented point cloud and concise piecewise-planar reconstruction (colored by plane / uncolored / wireframe contours).*

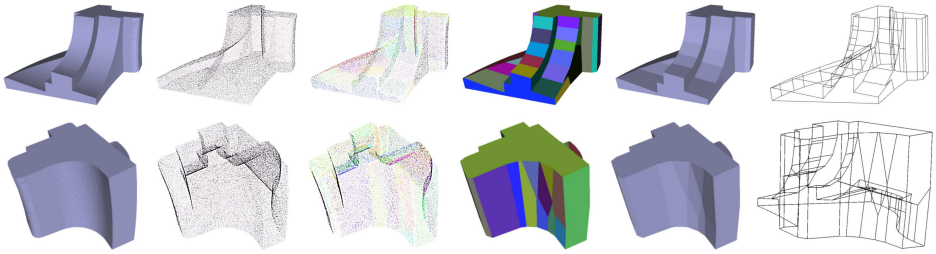
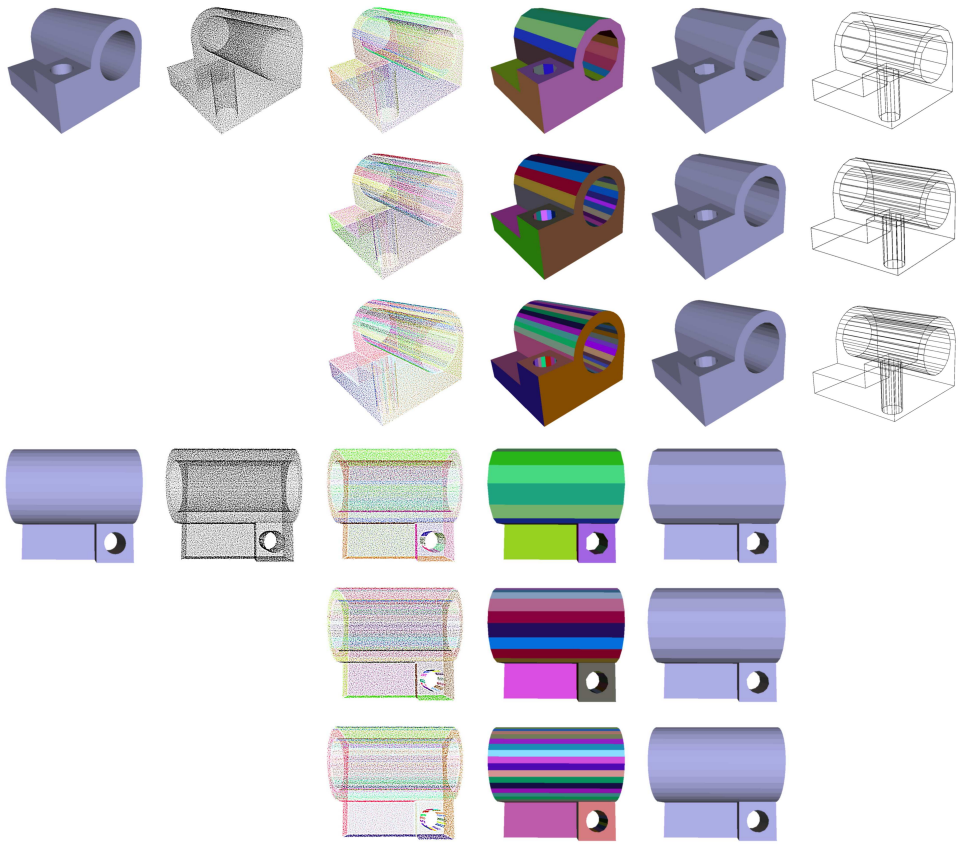


Figure 7.8: ***fandisk data set:*** *ground truth mesh, noisy input point cloud, segmented point cloud and concise piecewise-planar reconstruction (colored by plane / uncolored / wireframe contours).*

		cubes	*rue Soufflot*	*ALOI#418*	*cloister*	*castle-P30*
Input point cloud	#Points	110K	291K	58K	677K	888K
Segmentation	#Extracted planes	18	5	14	9	48
	#Clustered points	50K	199K	39K	194K	589K
		46%	68%	68%	29%	66%
Reconstruction	#Vertices (complex)	255	36	137	127	257
	#Facets (complex)	441	46	263	226	303
	#Vertices (surface)	48	18	44	48	160
		19%	50%	32%	38%	62%
	#Facets (surface)	36	10	36	38	95
		8%	22%	15%	16%	31%

Table 7.2: ***Detailed input and results:*** *data for the results on the other data sets.*

depicted by Figure 7.13. The object and its stand are properly reconstructed and the final model attempts to capture the simple geometry of this object.



		#Points	#Vertices	#Facets
	Original mesh		102K	204K
	Input point cloud	102K		
Reconstructions:	Concise reconstruction, $E(\mathcal{S}) = E_{vis}(\mathcal{S})$			
	$d_{max} = 10^{-2}$, $\theta_{max} = 22.5°$		139	112
	$d_{max} = 3.75\,10^{-3}$, $\theta_{max} = 7.5°$		203	155
	$d_{max} = 2.5\,10^{-3}$, $\theta_{max} = 6°$		260	198

Figure 7.9: ***joint data set:*** *different values of the distance threshold d_{max} and the normal deviation tolerance θ_{max} lead to varying levels of approximation of non piecewise-planar geometry.*

***cloister* (multi-view stereo)** This multi-view data set features 96 images of the cloister courtyard of the Cluny abbey. Our segmentation and reconstruction approach outputs a lightweight mesh featuring the ground, the facades and the roofs of this outdoor scene.

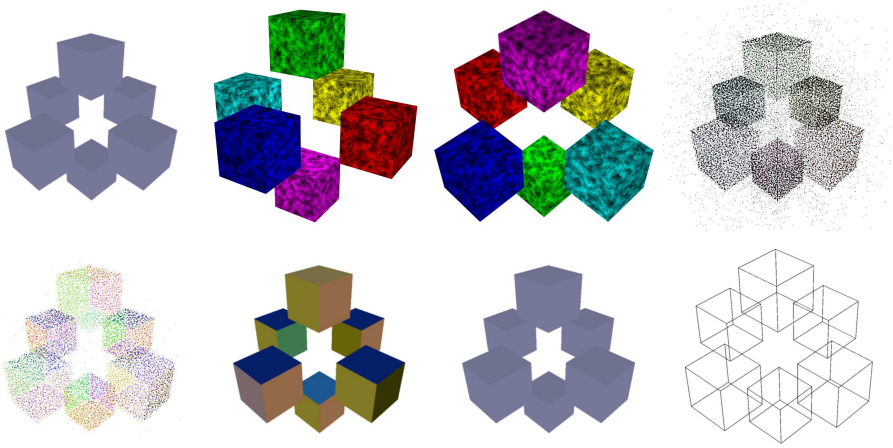
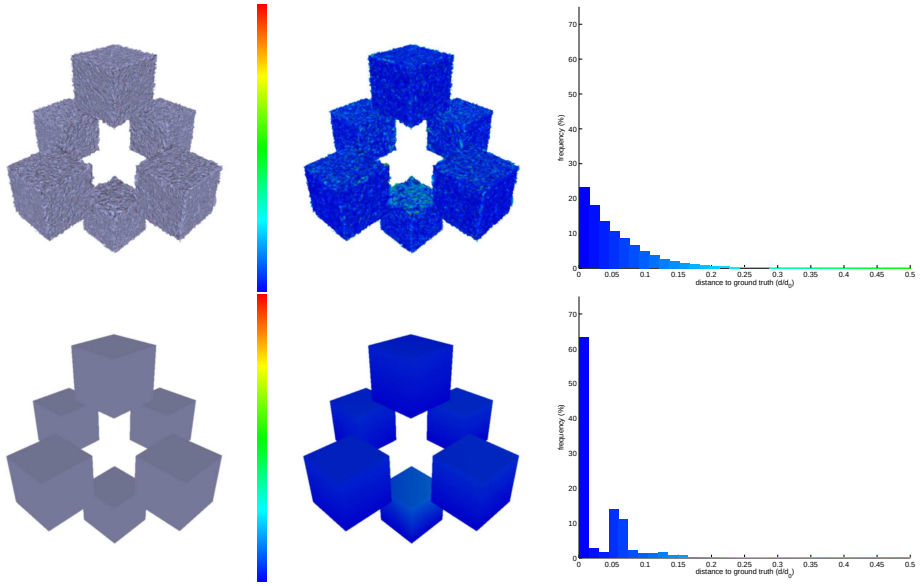


Figure 7.10: ***cubes data set:*** *a simple model, samples synthetic images, point cloud from interest points, segmented point cloud and concise piecewise-planar reconstruction (colored by plane / wireframe contours).*

***castle-P30* (multi-view stereo)** We finally consider the *castle-P30* data set from the recent dense multi-view stereo benchmark of large-scale scenes of [Strecha et al., 2008]. The 30 images (6 Mpix) of this dataset captures a courtyard containing large planar parts (the ground, the facades and the roofs) and some occluders (tractor, ...). Our result on a point cloud from the sparse depth maps of Chapter 1, shown in Figure 7.16, is the most impressive and preserves the large-scale structures of the courtyard. In Figure 7.15, The reconstruction algorithm of Chapter 4 failed to recover the ground due to the high number of outliers generated in the depth maps near the ground and the relatively low amount of inliers in this area. In contrast, our approach which search for large structures before reconstruction, is able to correctly recover the ground from inliers flooded in a mass of outliers and outputs a simple model that includes the main facades the ground, the roofs and the towers.



		#Points	#Vertices	#Facets
	Original mesh		48	36
	Input point cloud	110K		
Reconstructions:	Delaunay, $\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}}(\mathcal{S}) + \lambda_{\mathrm{qual}}\,\mathrm{E}_{\mathrm{qual}}(\mathcal{S})$		79K	159K
	Concise reconstruction, $\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}'}(\mathcal{S})$		48	36

Figure 7.11: ***cubes data set:*** *comparison with the generic reconstruction method of Chapter 3 (the normal estimate and outliers level were too high for Poisson to succeed). First row: Delaunay, second row: concise piecewise-planar reconstruction. First column: reconstructed surface, second column: reconstructed surface (colored by distance to ground truth), third column: histogram of distances to ground truth.*

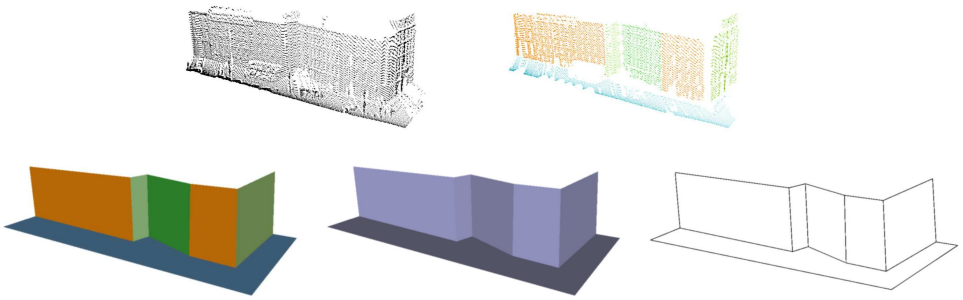


Figure 7.12: ***rue Soufflot data set:*** *a part of the previously shown* rue Soufflot *data set, segmented point cloud and concise piecewise-planar reconstruction (colored by plane / uncolored / wireframe contours).*

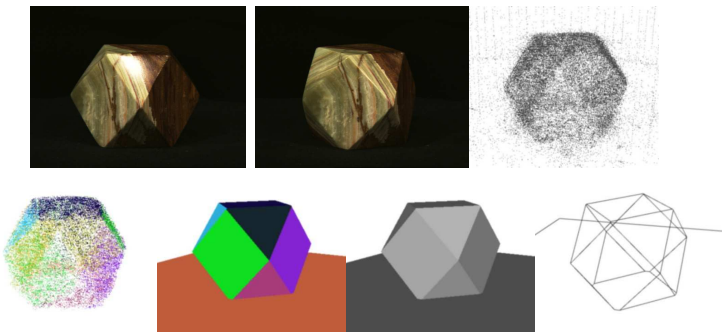


Figure 7.13: ***ALOI #418 data set:*** *two of the 72 input images, input point cloud, segmented point cloud and concise piecewise-planar reconstruction (colored by plane / uncolored / wireframe contours).*

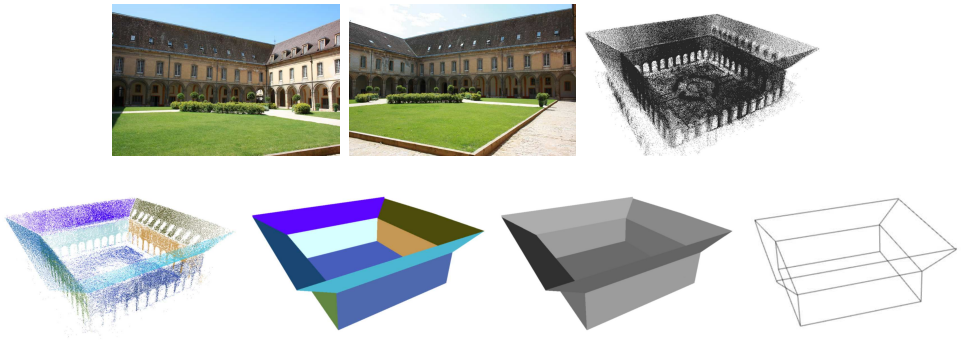


Figure 7.14: ***cloister data set:*** *two of the 96 input images, input point cloud, segmented point cloud and concise piecewise-planar reconstruction (colored by plane / uncolored / wireframe contours).*

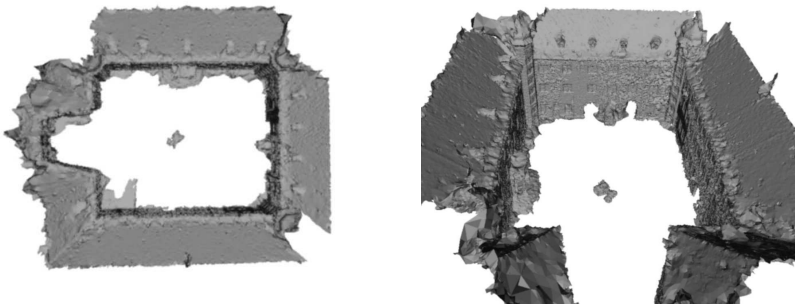


Figure 7.15: ***castle-P30 data set.*** *Initial mesh of the reconstruction method of Chapter 4 (top and front view): due to the high density of outliers, the ground is hardly reconstructed. The reconstruction has been smoothed for rendering purpose.*

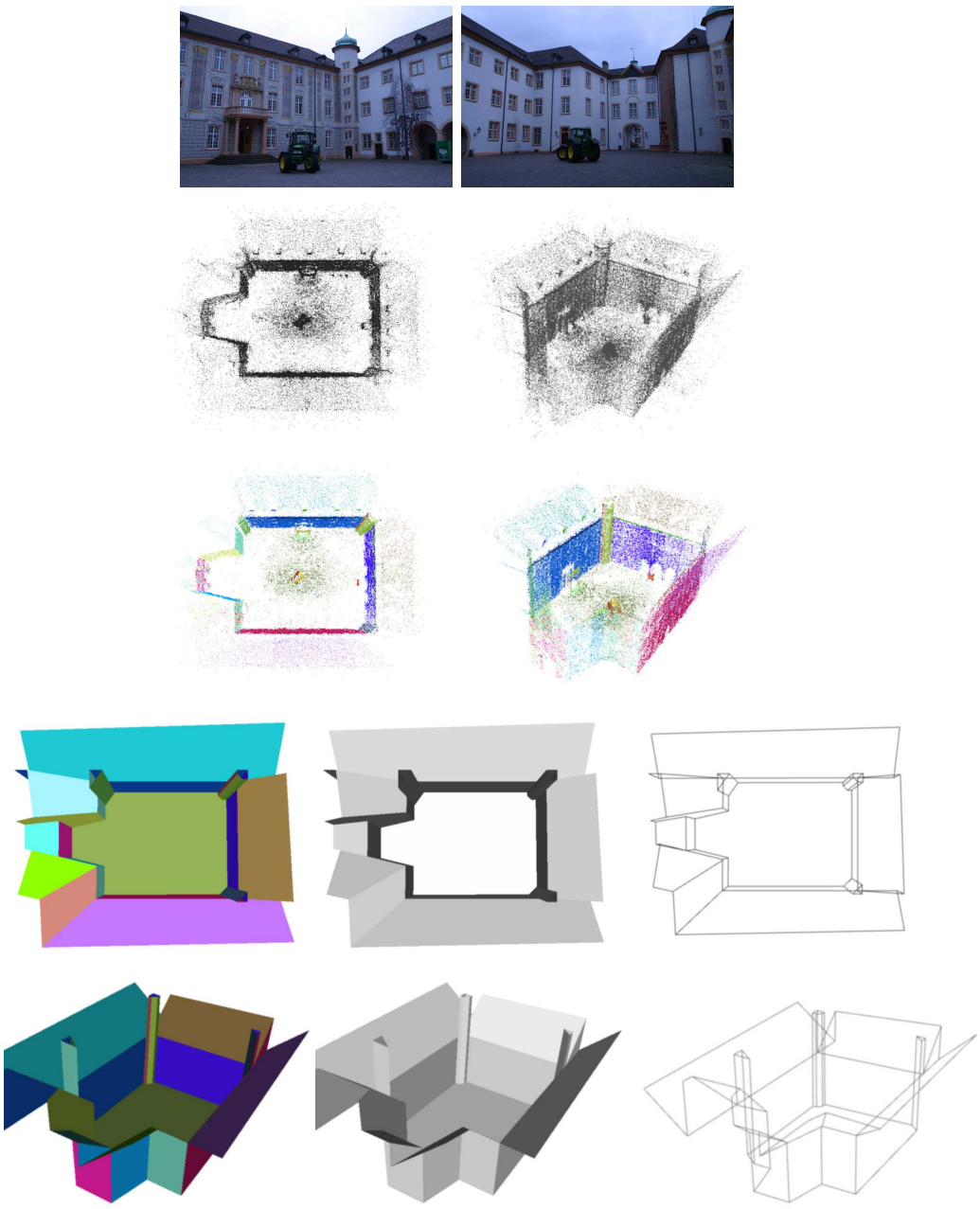


Figure 7.16: ***castle-P30 data set.*** *First row: two of the* 30 *images. Second row: point cloud from depth maps (top and side view). Third row: segmented point cloud. Fourth row: concise piecewise-planar reconstruction (colored by plane / uncolored / wireframe contours). Fifth row: colored concise reconstruction (front and back view).*

CHAPTER 8

Shape-based surface reconstruction

In contrast with Chapter 7, this chapter applies the hierarchical shape extraction method of Chapter 6 to several classes of shapes. The conciseness of the reconstructions is traded for the ability to model more complex surfaces. In addition, a simple extension is proposed to extract hybrid surfaces, which are not purely piecewise-primitive but can also interpolate through points not fitted to shapes. Encouraging results of the combined shape extraction and surface reconstruction approach are shown on both synthetic and challenging real data.

Contents

The second application of the hierarchical shape extraction procedure is extended to planes, spheres, cylinders and cones to produce a shape-based hierarchical description of the point set $\mathcal{P}$. As explained in Chapter 6, the fitting of second order surfaces is done *w.r.t.* the Euclidean distance to the shape. Details on the exact parametrization, initialization and fitting of shapes are given in Appendix A. The surface reconstruction uses as input, a subset $\mathcal{P}' \subseteq \mathcal{P}$ of the original input cloud along with the cluster representative $\mathcal{Sh}_P$ (the shape) to which each point P belongs and the tree description.

8.1 Surface reconstruction algorithm

The hierarchical segmentation induces a particular cell complex and with the notations introduced in Chapter 2, a pseudo-surface can be extracted corresponding to a globally optimum labeling of cells *w.r.t.* the following energy:

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}'}(\mathcal{S}, \mathcal{P}', v) \ ,$$

where $\mathrm{E}_{\mathrm{vis}'}$ is a variant of $\mathrm{E}_{\mathrm{vis}}$, similar to the one of the previous chapter, that penalizes surfaces crossing a line of sight and surface mis-orientations *w.r.t.* the clustered points $\mathcal{P}'$.

8.1.1 Generalized BSP complex and approximation

As in the previous chapter, the choice of cell complex is imposed by the hierarchical segmentation of the input point clouds into planes, spheres, cylinders and cones. Since the detected shapes are not restricted anymore to planes, the hierarchical structure is a generalized BSP tree. Each cell of the complex corresponds to a leaf of the BSP tree but one leaf of the tree may give rise to several cells (the space between two close parallel planes split by a sphere corresponds to two leaves but three cells for instance). Each facet of this complex is contained in one of the splitting surfaces. Two different cells may be linked by more than one facet since they may not be convex.

For polyhedral complexes induced by BSP complexes, the practical exact computation of complexes was possible. When the complex cells are delimited by general second order surfaces, this is still a subject of active research [Geismann et al., 2001, Fogel et al., 2006, Dupont et al., 2007], not to mention the queries also required by our visibility-based surface reconstruction framework. To circumvent this major problem, a practical solution is adopted: at the loss of conciseness, a refined Delaunay triangulation that approximates the shape supported facets is computed with a volume mesh generator derived from [Boissonnat and Oudot, 2005] from the BSP tree. A cell is approximated as a union of tetrahedra and a facet as union of triangles. A corresponding adjacency graph for cells is also recovered. Details for this step are given in Appendix B. The network graph may be a multi-graph

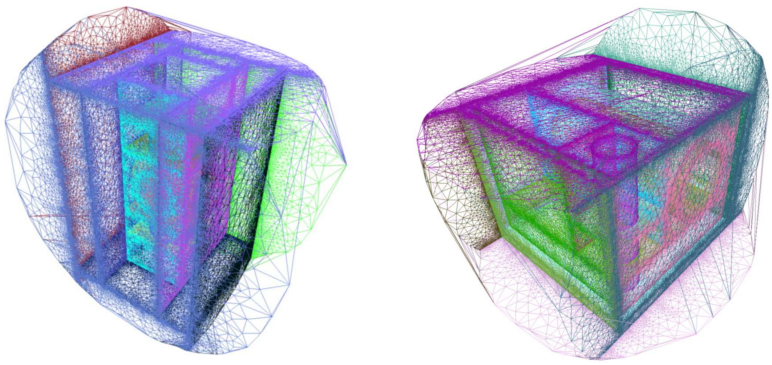


Figure 8.1: ***Generalized BSP complexes:*** *the final surface is extracted as a union of facets in a complex approximated by a triangulation. Left: complex for the* block *data set, right: for the* joint *data set.*

with several edges between the same pair of cells: these edges are merged together so that in the subsequent minimum s-t cut optimization the corresponding oriented facets are coupled. Examples of such approximated cell complexes are shown in Figure 8.1.

8.1.2 Surface visibility

The visibility term $\mathrm{E}_{\mathrm{vis}}$ of Chapter 2 is again slightly adapted to sets of points which are known to belong to clusters of points from an extracted shape in a fashion similar to the previous chapter. These fitted points are located near a facet of the complex whose supporting shape corresponds to their shape cluster as facets of the complex are built from shape clusters of points. The considered sequence of traversed cells by a line of sight is thus modified to account for this additional knowledge: the sequence stops at the facet $F \in \mathcal{C}_2$ of the complex corresponding to the shape $\mathcal{Sh}_P$ of the fitted point P and closest along the line of sight to it, provided such a facet exists. Using the same notations as in Chapter 2 with $N_{[QP]}$ the potentially

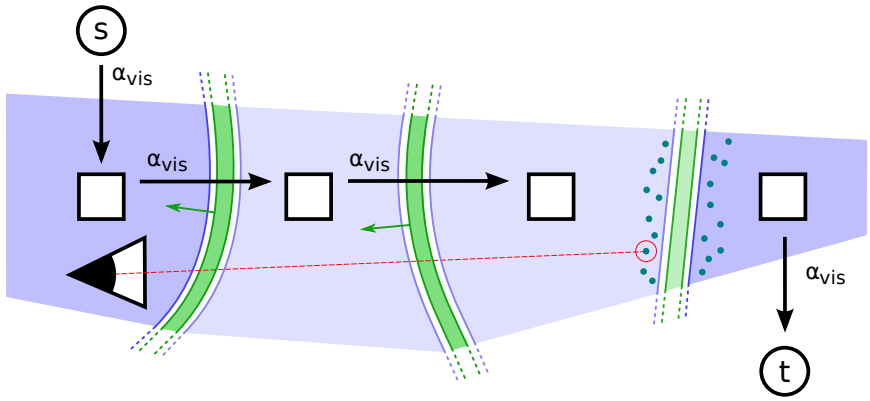


Figure 8.2: ***Visibility.*** *A fitted point (red circled dot) affects the oriented facets (darker green) and cells (darker blue) weights along one of its lines of sight (red dotted line).*

modified number of traversed cells in the sequence:

$$\mathrm{E}_{\mathrm{vis}'}(\mathcal{S},\mathcal{P}',v) = \sum_{P\in\mathcal{P}'} \sum_{\substack{Q\in v_P \\ \exists F\in C_2 \\ F\subset\mathcal{H}_P \wedge \dim([PQ]\cap F)=0 \\ \wedge F= \underset{\substack{G\in C_2 \\ G\subset\mathcal{H}_P \wedge \dim([PQ]\cap G)=0 \\ [PQ]\cap G=R}}{\arg\min} \left|\overrightarrow{PQ}.\overrightarrow{PR}\right|}} D_{\mathrm{out}}\left(l_{C_1^{Q\to P}}\right) + \sum_{i=1}^{N_{[QP]}-1} V_{\mathrm{align}}\left(l_{C_i^{Q\to P}}, l_{C_{i+1}^{Q\to P}}\right) + D_{\mathrm{in}}\left(l_{C_{N_{[QP]}+1}^{Q\to P}}\right) .$$

In terms of minimum s-t cut optimization and graph weights, just like in the previous chapter and as illustrated in Figure 8.2, the oriented facets crossed by the considered line of sight (darker green) get a weight of α_{vis}. The cell (darker blue) where the camera optical center lies is linked to the source s with an α_{vis} weight. Finally the cell behind the shape of the fitted point (darker blue) is linked to the sink t with a weight α_{vis}. These weights for cells being inside or outside the surface and facets being part of the reconstruction are accumulated for all the available lines of sight.

8.1.3 Hybrid surface reconstruction

As demonstrated in the previous chapter, a purely shaped-based reconstruction works well for scenes or for objects that can be easily decomposed or

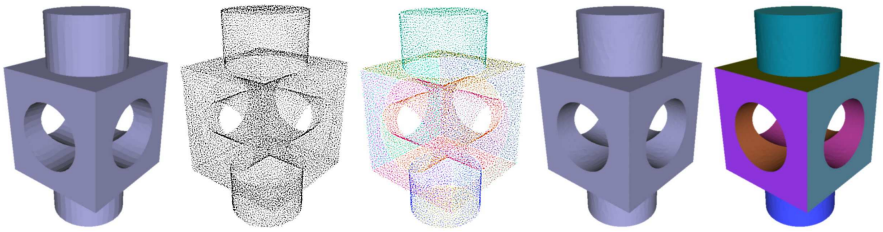


Figure 8.3: ***block data set:*** *ground truth mesh, noisy input point cloud, segmented point cloud and shape-based reconstruction (colored by shape / uncolored).*

approximated in surfaces patches from the set of shapes and when no severe occlusion hinders some parts of the scenes from being sampled by the point cloud. Since a purely shape-based reconstruction relies on a boundary representation and labels cells as inside or outside, missing surface patches required to close the object volume can lead to gross error in the reconstruction (see Figure 8.4), since the hole filling property of the Delaunay triangulation used in Chapters 3 and 4 is lost. Most of the geometry of the object can be captured by computing a hybrid reconstruction combining shape elements as before with some points of the depth maps to recover a faithful reconstruction of the whole scene. Similar hybrid meshes have been very recently proposed in a post-processing step [Lafarge et al., 2009] to be applied to the output of the dense multi-view stereo pipeline of Chapter 4. Here, the proposed hybrid surface reconstruction has the additional property to preserve sharp boundaries between neighboring shapes. This is achieved easily as follows: instead of restricting the optimization domain to the approximated BSP complex, the whole approximating triangulation is used, augmented with points that were not fitted to the detected shapes. This is done without altering the meshing of the shapes, by first computing the Delaunay triangulation of the points that were not fitted and then refining this triangulation (without modifying the vertices position) using the multi-domain mesh generator (and recovering approximated BSP complex cells and facets as above). This way, the optimization domain embeds both the shapes and the points likely to reside in fine or uncaptured details of the scene. The energy used in the optimization is modified as follows (the output surface is now a triangulated mesh whose facets are either unlabeled or labeled with their supporting shapes):

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}'}(\mathcal{S}, \mathcal{P}', v) + \mathrm{E}_{\mathrm{vis}}(\mathcal{S}, \mathcal{P} \backslash \mathcal{P}', v) + \lambda_{\mathrm{hybrid}}\, \mathrm{E}_{\mathrm{hybrid}}(\mathcal{S}) \ ,$$

where $\lambda_{\mathrm{hybrid}}$ is a positive weighting constant.

The term $\mathrm{E}_{\mathrm{vis}'}(\mathcal{S}, \mathcal{P}', v)$ penalizes visibility inconsistencies *w.r.t.* the fitted points and is the same term as previously but on the triangulated domain

instead: all triangular facets crossed by a line-of-sight are penalized and the sequence of traversed tetrahedra stops at the facet corresponding to the shape cluster of the considered point. The term $E_{vis}(\mathcal{S}, \mathcal{P}\backslash\mathcal{P}', v)$ refers to points that were not fitted to shapes and is exactly the same as the visibility term of Chapter 4 over Delaunay triangulations. Finally, the term $E_{hybrid}(\mathcal{S})$ penalizes only triangular facets that do not belong to facets of the BSP complex (*i.e.* , facets not belonging to recovered shapes) with the discrete quality cost of Chapter 3. As shown in Figure 8.4, adjusting the weight λ_{hybrid} allows switching from a purely shape-based reconstruction to a hybrid reconstruction with finer details (apart from Figure 8.4, all the presented hybrid reconstruction results use the same value of λ_{hybrid}).

Note that a similar approach might also be used in the purely piecewise-planar case to extend the approach of the previous chapter to better handle missing data which might occur in the case of serious occlusions: refine a Delaunay triangulations initially including the point not fitted and the fitted points projected to their respective planes so that the triangulation conforms to the facets of the polyhedral complex by adding Steiner points. Algorithms and implementation for this problem actually exists [Cohen-Steiner et al., 2002, Si and Gaertner, 2005, Si, 2007]. The drawback is however the same as here: the conciseness of the reconstructions of the previous chapter would be lost and a post-processing step that simplifies the mesh would be required.

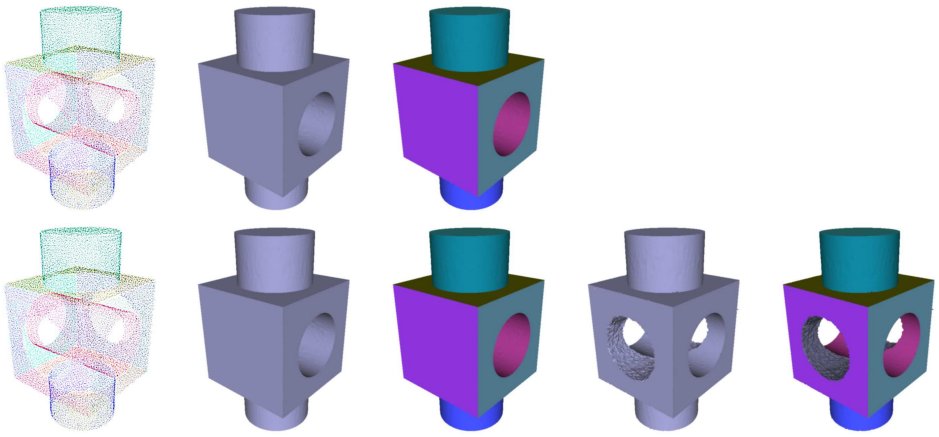


Figure 8.4: ***block data set.*** *First row: shape extraction result with one of the detected shape purposely removed and shape-based reconstruction result. Second row: hybrid reconstruction result with a high* λ_{qual} *and hybrid reconstruction result with a low* λ_{qual}.

8.2 Experimental results

Note that the results presented here use the variant of the extraction algorithm of Chapter 6 that only searches for shapes inside a cell and splits only that cell. The running time of the prototype implementation is more important than in the previous chapter due to the increased number of shape classes and the different cell complex construction: from 5m to about 1h for the extraction of shapes (the implementation can be improved and conservative settings were used for the random sampling optimization) and from 1m to 15m for the surface reconstruction.

Noisy point clouds (synthetic) Figures 8.3, 8.5 and 8.6 evaluate our shape-based segmentation and reconstruction pipeline on synthetic data. As in the previous chapters, the input point cloud was generated from the vertices of a mesh of the object. The associated visibility information was determined with ray-casting and occlusion computation using 64 virtual cameras around the object and some amount of isotropic Gaussian noise was added to the locations of the points. In these synthetic experiments, our purely shape-based reconstruction works very well and outputs a faithful representation of the original object. Note how the back of the shape-based reconstruction of the *fandisk* data set, which deviates slightly from cylinder is approximated with several cylindric patches. In Figures 8.7, 8.9 and 8.8, our reconstruction results on these data sets are compared with the surface reconstruction algorithm of Chapter 4 and Poisson surface reconstruction [Kazhdan et al., 2006]. The usual observations on these methods again apply here, [Kazhdan et al., 2006] smooth out sharp features and depends on consistent normal estimation (there is a small bump in the reconstruction of the *joint* data set due to an area with mis-oriented normals). The surface reconstruction algorithm from Chapter 4 is interpolatory and outputs noisy surfaces when applied to noisy inputs.

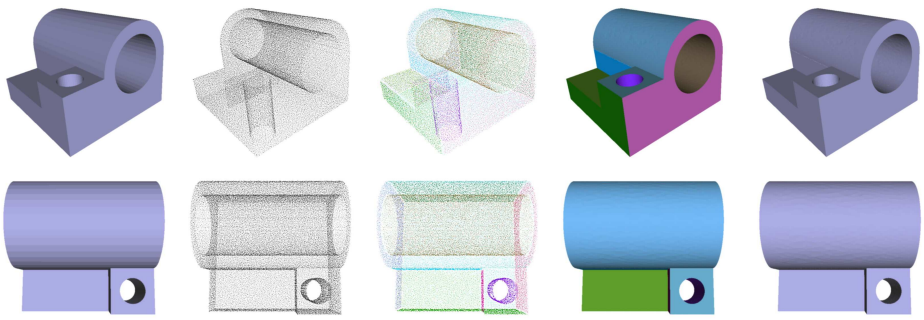


Figure 8.5: ***joint data set:*** *ground truth mesh, noisy input point cloud, segmented point cloud and shape-based reconstruction (colored by shape / uncolored).*

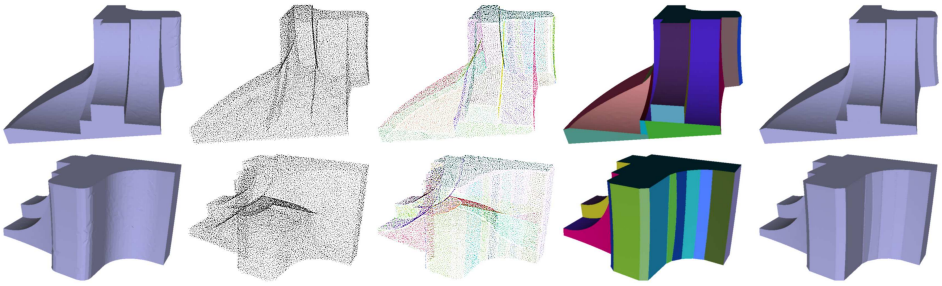


Figure 8.6: ***fandisk data set:*** *ground truth mesh, noisy input point cloud, segmented point cloud and shape-based reconstruction (colored by shape / uncolored).*

		block	*joint*	*fandisk*	*carter*	*castle-P30*	*fountain-P11*	*herzjesu-P25*
Input point cloud	#Points	49K	102K	44K	570K	888K	319K	718K
	σ_{noise}	0.07	0.0025	0.01				
	ℓ_{max}	38	1.12	5.25				
Segmentation	#Extracted shapes	12	12	32	40	14	63	139
	#Clustered points	46K	38K	77K	477K	440K	211K	364K

Table 8.1: ***Detailed input and results:*** *data for the results on the synthetic and real data sets. For the synthetic data sets, σ_{noise} is the standard deviation of the added isotropic Gaussian noise and ℓ_{max} is the length along its maximum dimension of the bounding box of the point cloud.*

***carter* (range data)** Figure 8.10 shows the result of our segmentation and reconstruction on a set of range scans of a mechanical part. While the structures making up the carter have been identified and reconstructed, due to the search and split strategy used in this chapter and the simplistic shape selection criterion, the inner and outer shell of the carter are jagged at the junction between different shapes.

***castle-P30* (multi-view stereo)** Our result is shown in Figure 8.11, and as expected, is a simplified model of the scene. The result is different from the previous chapter in terms of geometry, because a different distance threshold for fitting shapes was used. The combinatorial complexity of the resulting mesh is also much higher than for the lightweight result of the previous chapter.

***fountain-P11* (multi-view stereo)** The *fountain-P11* scene of Figure 8.12 combines easily identifiable shapes with very fine details making it suitable to challenge our hybrid surface reconstruction. The result of Chapter 4 in

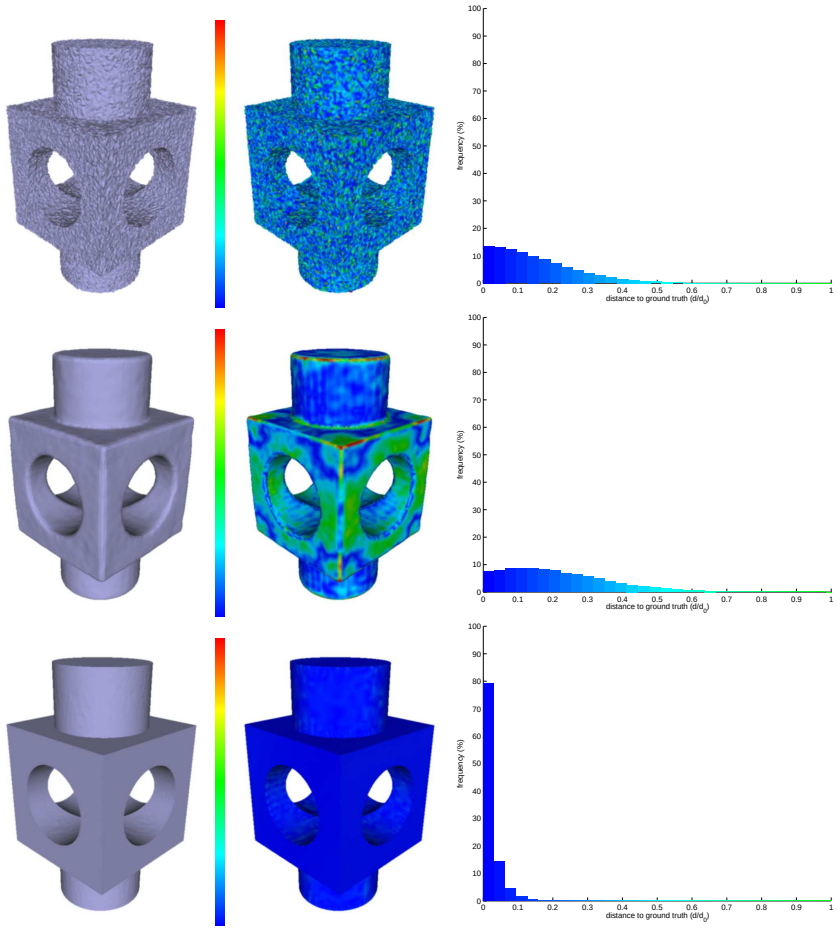


Figure 8.7: ***block data set:*** *comparison with the general purpose reconstruction methods of Chapter 4 and Poisson surface reconstruction. First row: Delaunay, second row: Poisson, third row: our shape-based reconstruction. First column: reconstructed surface, second column: reconstructed surface (colored by distance to ground truth), third column: histogram of distance to ground truth.*

Figure 8.12 is noisy but contains most of the details of the fountain. The hybrid reconstruction automatically extends the wall and ground, identifies the planar parts of the fountain base and approximates sculptures with small decorations with smooth second order patches.

***Herz-Jesu-P25* (multi-view stereo)** The results of Figure 8.13 on the *Herz-Jesu* scene are more striking. While the raw reconstruction of Chapter 4 is, as usual, a noisy model, the staircases of the hybrid reconstruction are perfectly straight (noisy staircases are typical cases of failure of RANSAC methods), the columns and archways are smooth and the facades and the

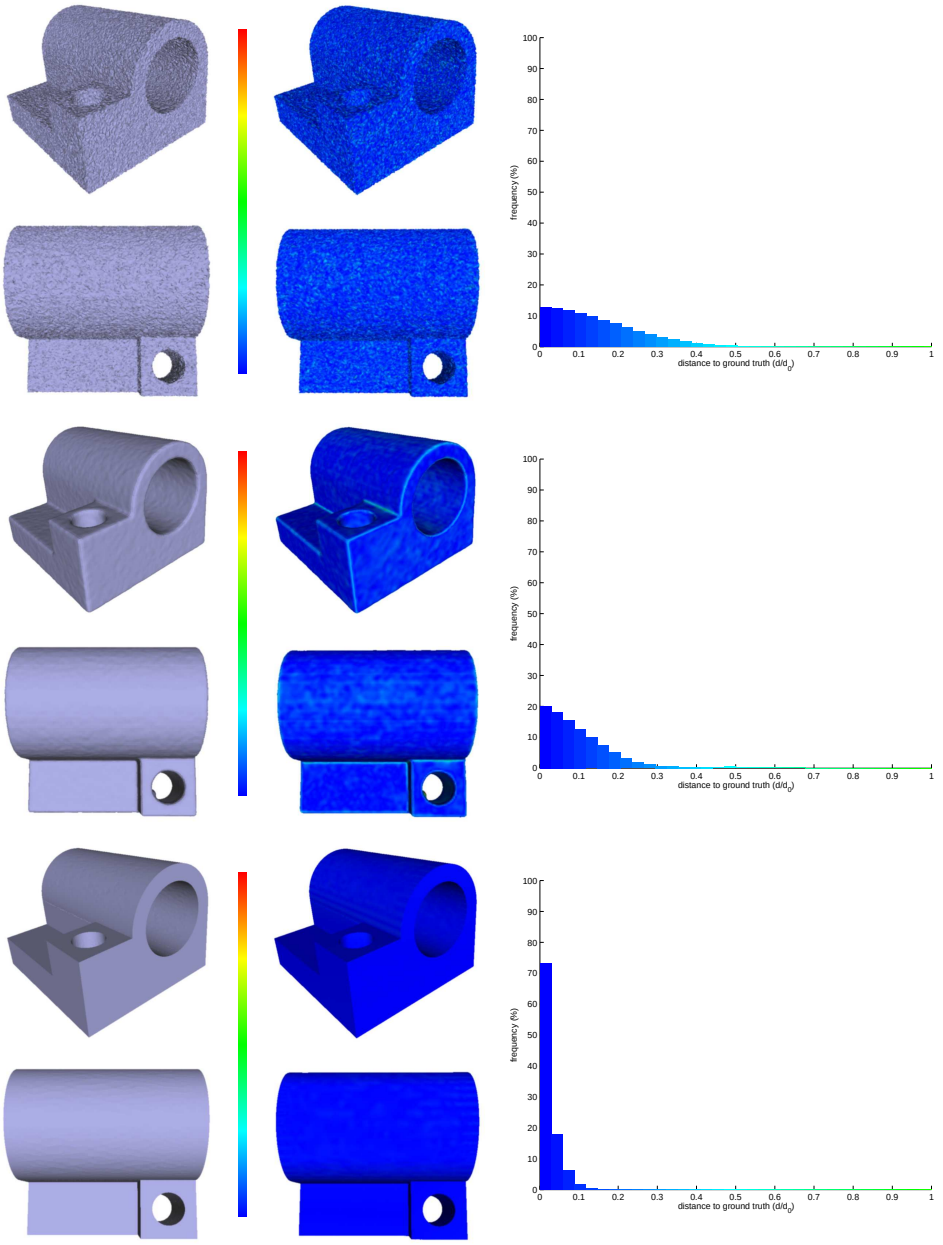


Figure 8.8: ***joint data set:*** *comparison with the general purpose reconstruction methods of Chapter 4 and Poisson surface reconstruction. First row: Delaunay, second row: Poisson, third row: our shape-based reconstruction. First column: reconstructed surface, second column: reconstructed surface (colored by distance to ground truth), third column: histogram of distance to ground truth.*

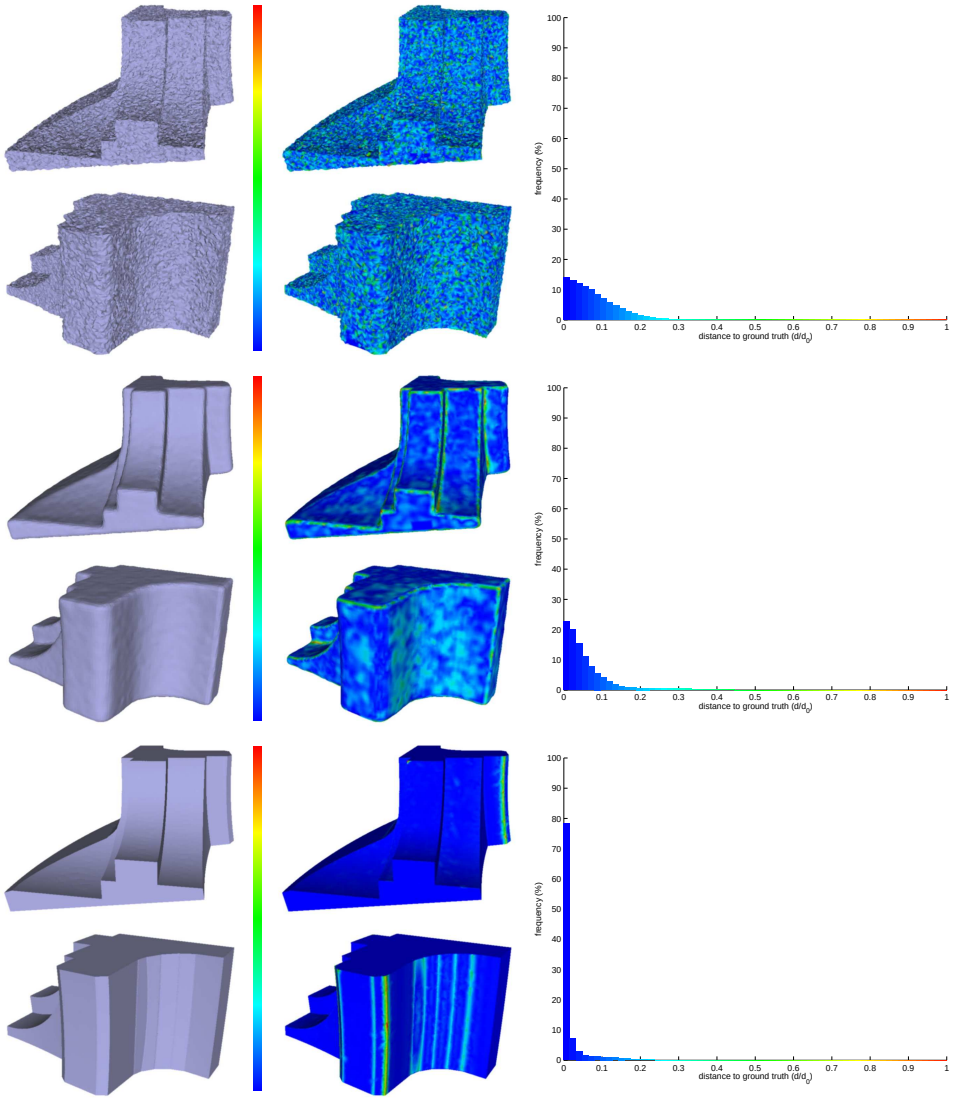


Figure 8.9: ***fandisk data set:*** *comparison with the general purpose reconstruction methods of Chapter 4 and Poisson surface reconstruction. First row: Delaunay, second row: Poisson, third row: our shape-based reconstruction. First column: reconstructed surface, second column: reconstructed surface (colored by distance to ground truth), third column: histogram of distance to ground truth.*

ground (which produces lots of mismatches) have been replaced by planes.

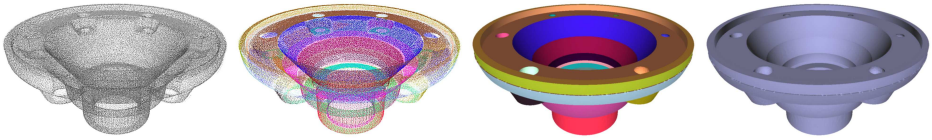


Figure 8.10: ***carter data set:*** *input point cloud, segmented point cloud, shape-based reconstruction (colored by shape / uncolored).*

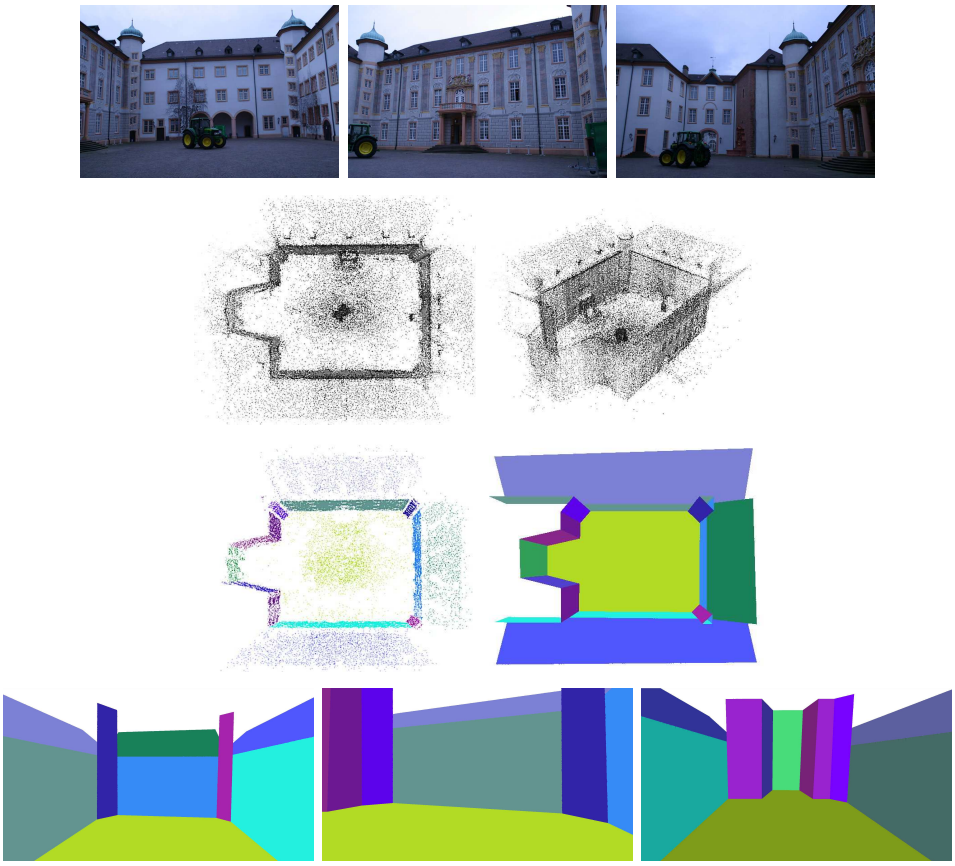


Figure 8.11: ***castle-P30 data set.*** *First row: three of the 30 images. Second row: point cloud from depth maps (top and side view). Third row: segmented point cloud and shape-based reconstruction (colored by shape). Fourth row: additional views of the shape-based reconstruction.*

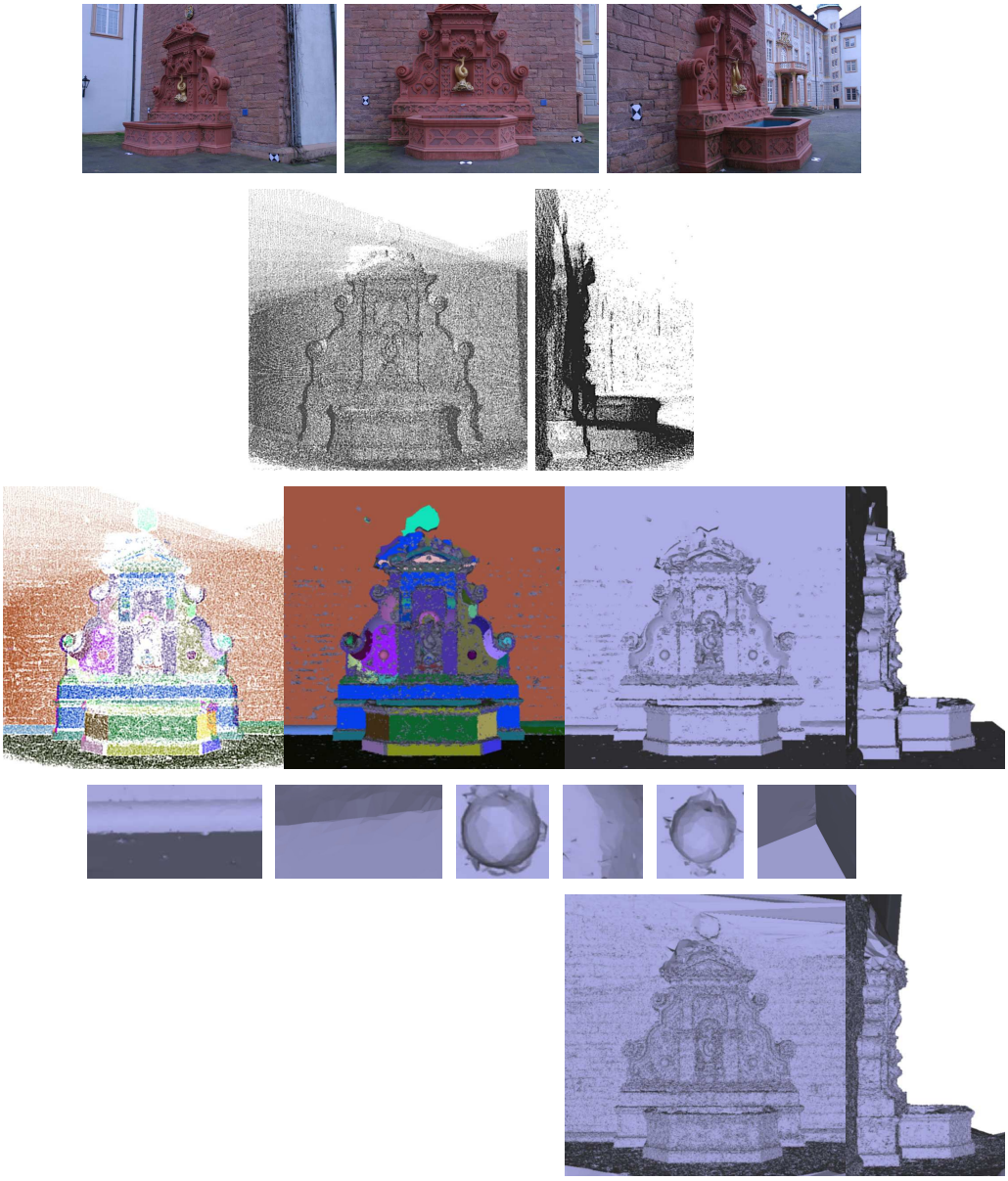


Figure 8.12: ***fountain-P11 data set.*** *First row: three of the 11 images. Second row: point cloud from depth maps (front and side view. Third row: segmented point cloud, hybrid reconstruction (colored by shape / uncolored from front and side view). Fourth row: close-ups on the hybrid reconstruction, from left to right: wall base and ground, basin side, decorations, basin interior. Fifth row: (unsmoothed) initial mesh of the dense multi-view stereo pipeline of Chapter 4.*

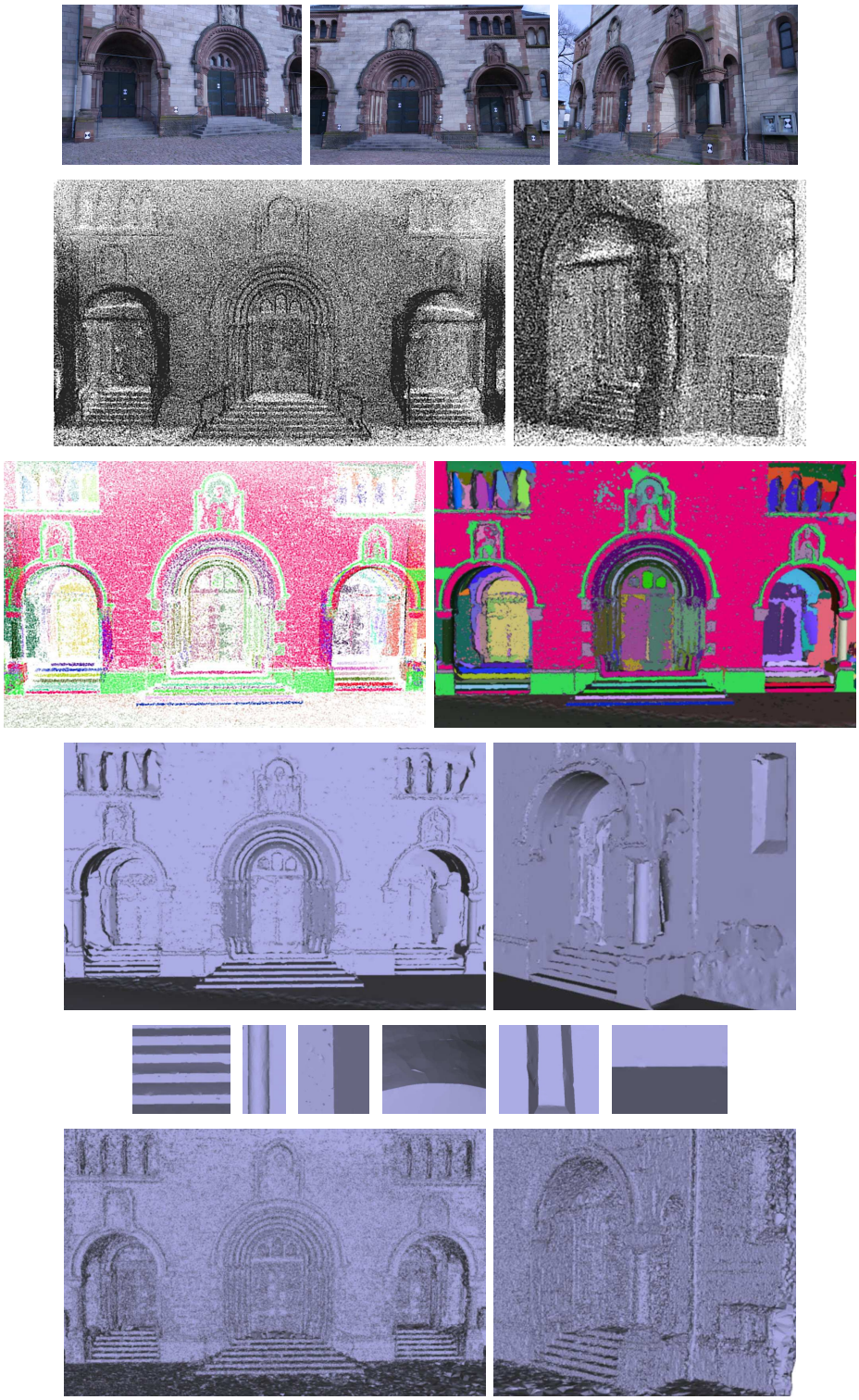


Figure 8.13: ***Herz-Jesu-P25 data set.*** *First row: three of the 25 images. Second row: point cloud from depth maps (front and side view). Third row: segmented point cloud and hybrid reconstruction colored by shape. Fourth row: hybrid reconstruction. Fifth row: close-ups on the hybrid reconstruction, from left to right: stairs, column, facade and side wall, archway, window, facade and ground. Sixth row: (unsmoothed) initial mesh of the dense multi-view stereo pipeline of Chapter 4.*

Conclusion

Summary

This thesis deals with surface reconstruction from point clouds acquired either with range-scanning devices or with simple image-based matching techniques for multi-view stereo.

A surface fitting framework has been proposed: the surface to reconstruct is sought as a surface that satisfies most visibility constraints coming from the points line-of-sights. The formulated optimization problem is solved efficiently and globally using minimum s-t cut. Compared to previous cut-based approaches, the surface fitting does not necessarily use a regular subdivision of the space to approximate integrals. The problem to be solved is instead purely discrete. Great flexibility comes from the use of a purely point-driven optimization domain, which allows the method to be applied to large outdoor scenes and also to handle closed as well as open scenes. Another major difference is the explicit avoidance of the "shrinking bias" that often plagues graph cuts approaches for minimal surfaces. The proposed framework is also versatile and extensible.

A first application is presented on the Delaunay triangulation of the input points to reconstruct a watertight triangular mesh from range images. Contrasting with other approaches, our method is simpler, being based on acquisition information. The original visibility term is relaxed to accommodate for noise in dense albeit noisy range scans and a discrete surface quality term is added. The robustness of the method is demonstrated on several examples: it is able to cope with severe undersampling, noisy data and outstanding amounts of outliers, and is applicable to large-scale outdoor scenes. While such conjunction of data alterations are rare for range scanner laboratory data, they are much more commonplace for outdoor range data acquisitions which justifies our approach and makes it a versatile tool for

surface reconstruction. In spite of a currently limited implementation, great computational scalability is achieved with respect to the size of the input point cloud both in running times and in memory use.

A second application of the surface fitting framework is found in multi-view stereo reconstruction of large-scale cluttered scenes taken under uncontrolled imaging conditions, a scenario where traditional multi-view stereo methods are either not applicable or have completeness and accuracy issues in part due to a lack of a correct treatment of visibility issues. From a quasi-dense point generated from the images, an initial surface is reconstructed by following the method used for reconstruction from range data: this surface is both complete and close to the ground truth and serves as a coarse initial estimate of the scene or object of interest. Its accuracy is then improved by a carefully designed and also scalable variational refinement. The complete multi-view stereo pipeline is demonstrated on various large-scale scenes. It outputs reconstructions that are visually and quantitatively more accurate and complete than state-of-the-art techniques.

Next, the problem of reconstructing compact models from point clouds or more generally surface reconstruction with shape priors has been considered. Our approach encompasses clustering, multiple structures detection in noisy point clouds with outliers and shape-based surface reconstruction. Several improvements to the random sampling scheme are proposed for dense and quasi-dense point clouds: the acquisition process is again relied upon to filter out outliers and the local density of the point cloud is exploited in the sampling and error measure. These modifications allow in practice the recovery of multiple structures in point clouds. The clustering of the input point cloud in different shapes (from predefined classes of shapes) is used in a subsequent reconstruction step that again exploits a variant of the visibility-based surface fitting framework.

When the set of extracted shapes is restricted to planes only, our final surface reconstruction is based on a polyhedral complex induced by the recovered planes and the resulting surfaces are extremely lightweight segmented mesh with polygonal facets that we call a “concise reconstruction”.

This conciseness is then traded for the recovery of more complex shapes, enabling piecewise-primitive reconstructions. The reconstruction method is finally extended to handle challenging point clouds of scenes that combines both fine details and easily identifiable shapes.

In both cases, encouraging results on synthetic and real-world data are shown and compared with reconstructions of general-purpose methods.

Limitations and perspectives

Despite the above achievements, the various approaches have provisions for future work.

As for our visibility-based surface fitting on point-driven cell complexes, a way to quantify the sampling requirements *w.r.t.* the particular chosen cell complex would help to make the reconstruction even more robust: the method is known to produce incorrect results in poorly sampled areas where outliers outweight inliers. Nevertheless, we believe that using line-of-sight or more generally exploiting as much knowledge as possible from the acquisition process, a recurring idea in this thesis, would make many graphics or vision related problem better posed.

The surface reconstruction algorithm from range data is interpolatory, which is actually one of its main limitations. A possible remedy would be to account for the noise model in the spatial subdivision, or perhaps more realistically, to split the visibility-based point filtering and surface reconstruction into two distinct steps. Another limitation is the scalability to very large data sets that are now common in range scanning [Levoy et al., 2000, Banno et al., 2008]. As noted in Chapter 3, previous reconstruction algorithms have been extended to out-of-core or streaming versions in order to handle massive data sets with limited memory [Fiorin et al., 2007, Allègre et al., 2007, Bolitho et al., 2007]. A first step in this direction would be the design of a selective visibility-based filtering of the input point cloud as in [Allègre et al., 2006] to eliminate redundant sample points in overly dense areas before applying the final reconstruction.

The multi-view stereo pipeline is clearly a first step towards high-resolution models that could compete with laser scans, but the road to them might not be that long. The availability of very high-resolution consumer-grade cameras will raise new issues like the problem of splitting the reconstruction problem into several smaller, more tractable ones [Zaharescu et al., 2008]. A growing source of data sets gaining interest from computer vision researchers is the community data available on websites such as Flickr or Google Images. This kind of data poses new challenges on scalability and robustness to dramatic appearance variability that some researchers have already tried to address in [Snavely et al., 2008, Goesele et al., 2007]. Another source of large image-based data sets for multiple view reconstruction comes from ground-based videos of cities acquired for services such as Google Street View. Impressive results have already been achieved in this area with the real-time reconstruction of urban scenes (processing a total of more than one million images) summarized in [Pollefeys et al., 2008]. The use of an cross-correlation score throughout the pipeline assumes the surface material is always close to Lambertian, an hypothesis not always verified especially in the new challenging data sets just mentioned. Extending the variational refinement to simultaneous estimation shape and radiance [Jin et al., 2003, 2005, 2008, Delaunoy et al., 2008] may increase robustness and also allow advanced appearance-based applications.

Several methods have been proposed to address the very challenging 4D stereo problem (multi-view stereo in time): space and motion carving [Vedula

et al., 2000], iterative reconstruction [Starck and Hilton, 2007], scene flow estimation [Vedula et al., 2005], variational methods [Pons et al., 2007a, Huguet and Devernay, 2007, Courchay et al., 2009], dynamic surface elements [Carceroni and Kutulakos, 2002], mesh-based models with fixed topology for tracking [Furukawa and Ponce, 2008, 2009]. In contrast with these methods, a variant of our surface fitting framework to sequences of points clouds [Aganj et al., 2009] would enable spatio-temporal reconstructions to be obtained at an acceptable computational cost without resorting to local optimization and relying on silhouettes or imposed models.

The surface reconstruction with shape priors is only a first attempt to solve a seemingly new problem. Preliminary results are encouraging but various improvements are possible and expected. The issue of automatic scale selection in the shape segmentation step could be resolved by adapting existing techniques [Fan and Pylvänäinen, 2008]. The error function in the random sampling scheme basically uses a simple weighted counting argument; also measuring the spread of points over the hypothesized shape would help to deal with highly non-uniform point clouds. The algorithm to extract multiple shapes is greedy: point fitted to a shape are never considered again. Although the method was found to behave robustly in the experiments, this step may benefit from a more global approach to clustering like [Toldo and Fusiello, 2008]. To allow our simplified modeling technique to better capture more general scenes, the shape detection should be extended to other classes of shapes: this is feasible but would gain from a better shape class selection criterion inspired by the minimum description length and minimum message length [Wallace and Boulton, 1968, Rissanen, 1978] concepts. Another limitation comes from the shape estimation: since least-squares fitting of the Euclidean distances is used, the underlying noise model is assumed to be isotropic Gaussian which as noted in Chapter 1 is far from reality especially in point clouds from passive stereo.

An important assumption of the shape extraction and reconstruction, not always fulfilled in practice, is the uniformity and density of the input point cloud: the concise and purely shape-based surface reconstruction step are in particular limited in robustness *w.r.t.* severe occlusions that prevent correct shape recovery. A pre-processing should detect and embed simplified shapes boundaries in the complex possibly using either the points cluster or the input images. Additionally, replacing the labeling of cells by a selection of facets of the complex would make the whole approach more practical without sacrificing the conciseness of the reconstruction in the piecewise-planar case. Our hybrid reconstructions are partly interpolatory and would require a final guided photometric refinement to be applied as a lightweight post-processing to locally improve fine details. Applying the shape-based pipeline to more data sets especially range scan data is also expected.

Finally, we plan to explore the generalization of our simplified modeling approach to possibly other problems: shape-based remeshing of noisy

polygon soups may be a first application.

Résumé détaillé de la thèse

Cette thèse étudie le problème de la reconstruction de surface à partir de nuages de points ou d'images multiples. Ce problème apparaît à la fois dans le domaine de l'imagerie numérique et de la vision par ordinateur et a toujours été l'objet d'une recherche active. En effet, la reconstruction automatique de modèles de scènes ou d'objets trouve de nombreuses applications dans divers domaines tels que la conception assistée par ordinateur, l'ingénierie à rebours ou le diagnostic, la thérapie et la chirurgie assistés par ordinateur. De tels modèles sont utiles en géologie et topographie pour la simulation numérique mais aussi en histoire et en art pour l'archivage digital de l'héritage culturel. Finalement l'industrie du divertissement est un client de choix pour ce type de techniques très demandées pour la conception de jeux video et de films réalistes.

Acquisition de nuages de points

Les nuages de points utilisés dans cette thèse sont issus de deux modalités d'acquisition bien différentes. On considère d'abord des nuages acquis par scanner laser 3D. Ce type de nuages est très dense, contient peu de mesures aberrantes et les mesures ont un niveau de bruit faible mais de l'ordre du pas d'échantillonnage. L'autre méthode d'acquisition considérée procède par stéréovision multivues passive. Cette méthode produit des nuages soit par appariement de points d'intérêt dans une paire d'images puis triangulation 3D, soit par balayage de plans pour produire des cartes de profondeurs non denses. Les nuages générés sont moins denses, beaucoup plus bruités et contiennent une bonne proportion de mesures aberrantes due à des erreurs d'appariement.

Dans les deux cas, l'espace compris entre le point 3D généré et la source ou le détecteur doit être vide si le point n'est pas une mesure aberrante: la

surface du solide ne peut pas intersecter le rayon ayant servi à l'acquisition du point correspondant.

Reconstruction de surface

Le problème de la reconstruction de surface à partir d'un nuage de points censé échantillonner un solide a suscité beaucoup de recherche et de nombreuses méthodes différentes existent. Malheureusement, ces méthodes sont le plus souvent applicables uniquement à des nuages de points d'excellente qualité comme ceux acquis à l'aide d'un scanner laser.

Tous les algorithmes de reconstruction de surface présentés dans cette thèse suivent la même approche:

1: Générer un ensemble de points $\mathcal{P}$ muni d'ensembles de visibilité v (pour chaque point, les sources ou détecteurs utilisés pour son acquisition),
2: Calculer un complexe cellulaire à partir de ces points,
3: Calculer un étiquetage binaire optimal $l^\star = \arg\min_l \mathrm{E}(l)$ des cellules du complexe de manière à minimiser une énergie traduisant des contraintes de visibilité (éventuellement accompagnée de termes complémentaires),
4: Extraire une pseudo-surface orientée $\mathcal{S}$ à partir de l'étiquetage intérieur/extérieur optimal $l^\star$.

L'hypothèse sous-jacente est que la surface recherchée peut être approchée par une union de facettes du complexe et que les points du nuages échantillonnent suffisamment densément les facettes de ce complexe.

Le système de contraintes de visibilité est le suivant. Le rayon depuis une source ou un détecteur vers un point acquis ne doit pas traverser la surface reconstruite. De plus l'espace situé derrière le point doit être à l'intérieur du solide et celui où se trouve la source ou le détecteur doit être à l'extérieur. On note $C_1^{Q\to P}, \ldots, C_{N_{[QP]}}^{Q\to P}$ la suite de $N_{[QP]}$ cellules traversées depuis la source ou le détecteur Q vers le point P considéré. On note $C_{N_{[QP]}+1}^{Q\to P}$ la cellule situé derrière la facette la plus proche du point P dans cette direction. Les contraintes de visibilité se traduisent en une énergie $\mathrm{E}_{\text{vis}}(\mathcal{S}, \mathcal{P}, v)$ à minimiser :

$$\begin{aligned}\mathrm{E}_{\text{vis}}(\mathcal{S}, \mathcal{P}, v) = \mathrm{E}_{\text{vis}}(l, \mathcal{P}, v) = \sum_{P\in\mathcal{P}} \sum_{Q\in v_P} & D_{\text{out}}\left(l_{C_1^{Q\to P}}\right) \\ & + \sum_{i=1}^{N_{[QP]}-1} V_{\text{align}}\left(l_{C_i^{Q\to P}}, l_{C_{i+1}^{Q\to P}}\right) \\ & + D_{\text{in}}\left(l_{C_{N_{[QP]}+1}^{Q\to P}}\right),\end{aligned}$$

V_{align} étant un terme d'interaction binaire défini pour deux cellules adjacentes du complexe :

$$\begin{aligned} V_{\text{align}}\left(l_{C_i}, l_{C_j}\right) &= \alpha_{\text{vis}}\, \mathbb{1}\left[\, l_{C_i} = 0 \wedge l_{C_j} = 1 \,\right] , \\ D_{\text{out}}\left(l_C\right) &= \alpha_{\text{vis}}\, \mathbb{1}\left[\, l_C = 1 \,\right] , \\ D_{\text{in}}\left(l_C\right) &= \alpha_{\text{vis}}\, \mathbb{1}\left[\, l_C = 0 \,\right] , \end{aligned}$$

où α_{vis} reflète la confiance attribuée au point considéré.

Cette énergie peut être minimisée globalement en calculant une coupe minimale s-t sur un graphe dérivé du graphe d'adjacence des cellules du complexe. Un étiquetage intérieur/extérieur optimal des cellules du complexe peut donc être obtenu efficacement ainsi que la pseudo-surface orientée associée.

Reconstruction de surface à partir d'acquisitions laser

Notre approche est appliquée une première fois pour la reconstruction de surface à partir d'acquisitions laser en choisissant comme complexe cellulaire la triangulation de Delaunay des points du nuage. L'énergie minimisée est la suivante :

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\text{soft}-\text{vis}}(\mathcal{S}, \mathcal{P}, v) + \lambda_{\text{qual}}\, \mathrm{E}_{\text{qual}}(\mathcal{S}) \, .$$

Le terme $\mathrm{E}_{\text{soft}-\text{vis}}(\mathcal{S}, \mathcal{P}, v)$ est une variante relaxée des contraintes de visibilité. C'est une somme de pénalités pour la mauvaise orientation et le mauvais alignement de la surface $\mathcal{S}$ par rapport aux rayons issus des points vers les sources d'acquisition. Le terme $\mathrm{E}_{\text{qual}}(\mathcal{S})$ pénalise les triangles de $\mathcal{S}$ n'appartenant vraisemblablement pas à la surface recherchée. λ_{qual} est une constant positive d'ajustement.

Plusieurs expériences sont proposées afin d'évaluer la robustesse de notre algorithme au sous-échantillonnage, au bruit et aux mesures aberrantes en comparaison avec d'autres algorithmes existants. Enfin, une comparaison avec l'état de l'art [Kazhdan et al., 2006] est présentée sur un nuage de points de grande échelle acquis en extérieur depuis un véhicule. Notre algorithme passe bien à l'échelle en temps de calcul et en mémoire et se montre très robuste aux différentes dégradations des nuages en entrée.

Reconstruction dense multivues de scènes de grande échelle

Malgré des résultats de plus en plus précis [Seitz et al., 2006], la plupart des approches actuelles pour la photo-modélisation précise sont inadaptées à la reconstruction de scènes extérieures de grande échelle, soit parce qu'une enveloppe visuelle est nécessaire, soit parce que l'outil de minimisation repose sur une subdivision fine de l'espace ambiant.

Une variante de la méthode de reconstruction à partir de données laser est utilisée pour initialiser un raffinement photométrique variationnel. Puisque l'initialisation est déjà proche de la surface recherchée, la descente de gradient correspondante ne devrait pas tomber pas dans un minimum local.

L'énergie servant à étiqueter les tétrahèdres de la triangulation de Delaunay est la suivante :

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}}(\mathcal{S}, \mathcal{P}, v) + \lambda_{\mathrm{qual}}\, \mathrm{E}_{\mathrm{qual}}(\mathcal{S}) \ ,$$

où $\mathcal{P}$ est le nuage de points généré pas stéréovision multivues passive et v les ensembles de visibilité associés. La surface obtenue par cette première minimisation est notée $\mathcal{S}^0$.

L'énergie minimisée par le raffinement variationnel est :

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{error}}(\mathcal{S}) + \lambda_{\mathrm{thin-plate}}\, \mathrm{E}_{\mathrm{thin-plate}}(\mathcal{S}) \ ,$$

où $\mathrm{E}_{\mathrm{error}}$ mesure, comme dans [Pons et al., 2007a], l'erreur de reprojection de la surface $\mathcal{S}$ (liée à la photo-consistance de la surface dans les images) et $\mathrm{E}_{\mathrm{thin-plate}}$ pénalise les zones de forte courbure. La surface $\mathcal{S}$ suit alors la descente de gradient suivante :

$$\mathcal{S}(0) = \mathcal{S}^0 \qquad \frac{\mathrm{d}\mathcal{S}}{\mathrm{d}t} = -\nabla \mathrm{E}(\mathcal{S}) \ .$$

La surface $\mathcal{S}$ est représentée par un maillage triangulé pour faciliter le calcul du terme de photo-consistance et permettre une reconstruction précise tout en étant capable de passer à l'échelle. La descente de gradient employée est cohérente avec la représentation discrétisée de la surface.

Enfin, le banc d'essai de [Strecha et al., 2008] est utilisé pour démontrer la supériorité de cette chaîne de photo-modélisation par rapport aux autres méthodes pouvant reconstruire des scènes extérieures de grande échelle. Des résultats qualitatifs sur d'autres jeux de données sont également présentés.

Reconstruction de modèles compacts

Nous dressons d'abord un bilan des différentes approches applicables pour la reconstruction de modèles compacts à partir d'images ou de données laser. Que ce soit en vision par ordinateur ou en imagerie numérique, il n'existe pas de méthode satisfaisante permettant de reconstruire directement de tels modèles.

Nous proposons d'appliquer notre approche pour la reconstruction de surface sur un complexe cellulaire particulier. Des primitives issues d'un ensemble restreint de classes de primitives sont détectées de manière hiérarchique dans le nuage de points. Le complexe cellulaire induit par cette partition binaire de l'espace est utilisé comme domaine d'optimisation pour extraire une surface « primitive par morceaux ».

Extraction de primitives

Partant du constat que les nombreuses méthodes existantes pour l'extraction de primitives sont inadaptées soit à la détection de structures multiples, soit à la présence de bruit et de points aberrants, plusieurs améliorations au schéma classique RANSAC [Fischler and Bolles, 1981] sont proposées pour des nuages denses ou quasi-denses: le processus d'acquisition est pris en compte encore une fois et la densité locale de points non aberrants est utilisée dans la mesure d'erreur et l'échantillonnage des primitives. Ces modifications permettent en pratique la détection de plusieurs structures dans les nuages de points considérés dans cette thèse. Cette détection robuste de primitives est intégrée dans la décomposition hiérarchique de l'espace nécessaire pour appliquer notre approche pour la reconstruction de surface.

Reconstruction concise de surface planaire par morceaux

Lorsque que l'ensemble des classes de primitives est restreint aux plans seulement, le complexe cellulaire est un complexe polyhédral. Ce complexe peut être construit en utilisant une simple variante d'un algorithme classique [Edelsbrunner, 1987] pour la construction incrémentale d'arrangements de plans. Les reconstructions obtenues sur des nuages synthétiques et réels sont bien meilleures que celles obtenues par des méthodes génériques et les maillages polygonaux segmentés résultants ont une complexité très faible justifiant l'appellation de « reconstruction concise ».

Reconstruction de surface primitive par morceaux

Pour la détection et la reconstruction de surface à l'aide de primitives plus élaborées (plans, cylindres, sphères et cônes), la construction exacte du complexe cellulaire induit par les primitives détectées dans le nuage de point est abandonnée au profit d'un calcul approché. À cet effet, le mailleur multi-domaines de [Pons et al., 2007b] est utilisé pour obtenir une triangulation labélisée de l'espace. À chaque cellule correspond une union de tétrahèdres et à chaque facette, une union de triangles. La reconstruction de modèles compacts est illustrées sur plusieurs résultats synthétiques et réels. Même si le modèle produit par l'algorithme n'a plus la propriété de faible complexité des reconstructions concises, il est dorénavant possible de produire des modèles hybrides interpolant des points du nuage initial n'appartenant pas à des primitives détectées.

Enfin, notons qu'une partie du contenu de cette thèse provient d'articles publiés, à paraître ou en préparation:

- le système de contraintes de visibilité a été initialement introduit pour des triangulations de Delaunay dans l'article *Efficient Multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and graph cuts* [Labatut et al., 2007] avec Jean-Philippe Pons et Renaud Keriven dans les actes de l'IEEE International Conference on Computer Vision en 2007;
- l'algorithme de reconstruction de surface à partir de données laser est décrit dans l'article de journal *Robust and efficient surface reconstruction from range data* [Labatut et al., 2009a] à paraître dans Computer Graphics Forum avec Jean-Philippe Pons et Renaud Keriven;
- la chaîne complète de photo-modélisation est un travail commun avec Jean-Philippe Pons, Renaud Keriven et Vu Hoang Hiep et apparaît dans *Towards high-resolution large-scale multi-view stereo* [Vu et al., 2009] avec Vu Hoang Hiep, Renaud Keriven et Jean-Philippe Pons dans les actes de l'IEEE Conference on Computer Vision and Pattern Recognition, 2009. Un article de journal intitulé *Global and visibility-consistent dense multi-view stereo for large-scale scenes* avec Jean-Philippe Pons, Renaud Keriven et Vu Hoang Hiep, décrivant la chaîne de manière plus détaillée est en préparation;
- la méthode d'extraction hiérarchique de primitives et de reconstruction de surface est tirée de l'article *Hierarchical shape-based surface reconstruction for dense multi-view stereo* [Labatut et al., 2009b] avec Jean-Philippe Pons et Renaud Keriven à paraître dans les actes du 2009 IEEE International Workshop on 3-D Digital Imaging and Modeling.

APPENDIX A

Parameterization and fitting of shapes

This chapter presents the various parametrizations, initializations, distances formulae and optimization problems[a] for fitting shapes to sets of points used in the shape extraction algorithm of Chapter 6 and in the results presented in Chapters 7 and 8. We mostly follow [Shakarji, 1998] for the shapes parametrizations and their respective fitting optimizations.

[a] While the plane fitting case is pretty well-known and is linked to principal component analysis, it is nevertheless included for completeness.

Contents

A.1 Least-squares fitting

"Fitting" consists in adjusting the parameters $\boldsymbol{\beta} = (\beta_1, \dots \beta_n)^T \in \mathbb{R}^n$ of a function $f(., \boldsymbol{\beta}) : \mathbb{R}^m \to \mathbb{R}$ to best approximate a set of N data measurements $(\boldsymbol{x}_i, y_i)_{i \in [\![1,N]\!]}$. A data measurement is a pair $(\boldsymbol{x}, y)$ where $\boldsymbol{x} \in \mathbb{R}^m$ is an independent variable and $y \in \mathbb{R}$ a dependent variable found by observation. The least squares method, which is the simplest and most commonly used form of fitting, finds parameters $\boldsymbol{\beta}^\star$ (and a corresponding function) that minimizes the sum J of the squared N residuals $r_i(\boldsymbol{\beta}) = y_i - f(\boldsymbol{x}_i, \boldsymbol{\beta})$:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{N} r_i(\boldsymbol{\beta})^2 = \sum_{i=1}^{N} \left(y_i - f(\boldsymbol{x}_i, \boldsymbol{\beta})\right)^2$$

Or more compactly, with $R(\boldsymbol{\beta}) = \big(r_i(\boldsymbol{\beta})\big)^T$, $Y = \big(y_i\big)^T$ and $F(\boldsymbol{\beta}) = \big(f(\boldsymbol{x}_i, \boldsymbol{\beta})\big)^T$:

$$S(\boldsymbol{\beta}) = \|R(\boldsymbol{\beta})\|^2 = \|Y - F(\boldsymbol{\beta})\|^2$$

Assuming the measurements y_i have errors independently random and distributed as a normal distribution $\mathcal{N}(0, \sigma^2)$ around the true model, least-squares fitting is a maximum likelihood estimator since the likelihood $L(Y \,|\, \boldsymbol{\beta})$ function is:

$$L(Y \,|\, \boldsymbol{\beta}) = \prod_{i=1}^{N} P(y_i \,|\, \boldsymbol{\beta}) \propto \exp\left(-\frac{\sum_{i=1}^{N} r_i(\boldsymbol{\beta})^2}{\sigma^2}\right)$$

Weighted least-squares fitting is a variant that weights each residuals r_i with a corresponding coefficient $w_i > 0$. For the particular problem of fitting shapes to a set of points, the measurements are $(\boldsymbol{x}_i, 0)$ (where $\boldsymbol{x}_i \in \mathbb{R}^3$ are the data points), the function f is simply the (orthogonal) distance of a point to a shape with parameters $\boldsymbol{\beta}$, and finally, the weights are problem-specific confidences in the data points.

Linear least-squares

When f depends linearly on its parameters, one has $f(\boldsymbol{x}, \boldsymbol{\beta}) = \sum_{j=1}^{n} \beta_j \, \phi_j(\boldsymbol{x})$ (the ϕ_j being functions from $\mathbb{R}^m$ to $\mathbb{R}$), or equivalently $F(\boldsymbol{\beta}) = X\boldsymbol{\beta}$ with $X = \big(\phi_j(\boldsymbol{x}_i)\big) \in \mathcal{M}_{N,n}(\mathbb{R})$. The optimization problem then reduces to finding the vector $\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \|Y - X\boldsymbol{\beta}\|^2$ which leads to the so-called *normal equations* $X^T X \boldsymbol{\beta}^\star = X^T Y$. These equations are typically solved using a Cholesky decomposition of $X^T X$, a QR or an SVD decomposition of X by increasing order of stability *w.r.t.* a potentially ill-conditioned $X^T X$.

Non-linear least-squares

More generally, f may non-linearly depend on its parameters $\boldsymbol{\beta}$. This is the case of interest here where a shape is fitted by minimizing the orthogonal distance of a set of points. Only first-order surfaces (*i.e.* , planes) least-squares fitting reduces to a simple problem (a 3×3 eigenvalue problem). Second-order surface fitting we are also interested in require the non-linear least-squares problem to be solved iteratively with an appropriate initialization:

$$\begin{cases} \boldsymbol{\beta}_0 \\ \boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \Delta\boldsymbol{\beta} \end{cases}$$

To find the increment $\Delta\boldsymbol{\beta}$, a simple gradient descent can be used: $\Delta\boldsymbol{\beta} = \gamma J^T R$ (noting J the Jacobian matrix of F and γ a small enough positive constant). While this approach is guaranteed to converge to a local minimum, it is usually very slow. Using a Taylor expansion of the residuals, $S(\boldsymbol{\beta}_{k+1}) = \|Y - F(\boldsymbol{\beta}_k + \Delta\boldsymbol{\beta})\|^2 \approx \|Y - F(\boldsymbol{\beta}_k) - J\Delta\boldsymbol{\beta}\|^2$ is now a linear least-squares problem in $\Delta\boldsymbol{\beta}$ leading to the Gauss-Newton method based on the corresponding normal equations: $J^T J\Delta\boldsymbol{\beta} = J^T R$. This method is faster with convergence approaching a quadratic rate, however it is not guaranteed to converge and is even known to fail when starting far off the final minimum. The well-known Levenberg-Marquardt algorithm [Levenberg, 1944, Marquardt, 1963] "interpolates" between these two methods (the components of the gradient are additionally scaled to improve the convergence):

$$\left(J^T J + \lambda \operatorname{Diag}(J^T J)\right) \Delta\boldsymbol{\beta} = J^T R$$

where $\lambda \geq 0$ is a "damping" parameter. It is one of the most popular method for least-squares curve and surface fitting. While Levenberg-Marquardt is slower than Gauss-Newton, in practice, it is more robust and able to reach a solution even when started far off from the final minimum.

The rest of this chapter provides, for each considered class of shape, its parametrization $\boldsymbol{\beta}$, the associated distance formula, the initialization from a minimal set of points, the objective function $S(\boldsymbol{\beta})$ and the coefficients of the Jacobian matrix $J(\boldsymbol{\beta})$. A modification to the general procedure of the Levenberg-Marquardt algorithm is the normalization of the parameters of each shape applied at the beginning of each iteration before computing the distances and Jacobians of their respective least-squares problems. This normalization does not affect the optimization process. The actual implementation of the method described in Chapter 6 relies on the MINPACK [Moré et al., 1980, 1984, min, 1999] library of routines. MINPACK provides highly portable and robust implementations of several algorithms for solving non linear systems and minimization problems. In particular the `lmdif` and `lmder` routines are implementations of the Levenberg-Marquardt algorithm

with automatic choice of damping parameter. More specifically the C/C++ port of MINPACK of [Devernay, 2009] was used.

A.2 Plane

Parametrization

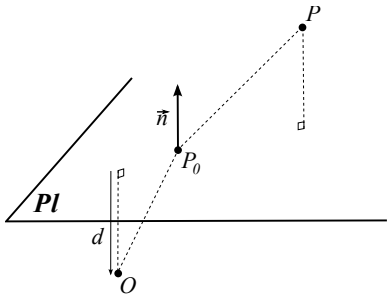


Figure A.1: *A plane $\mathcal{Pl}_{\vec{n},d}$, the geometric meaning of its parameters and a point P.*

A plane $\mathcal{Pl}_{\overrightarrow{n},d}$ is parametrized by $\overrightarrow{n}$ and d where (see Figure A.1):

- $\overrightarrow{n} = (n_x, n_y, n_z) \in \mathbb{R}^3$ is a vector normal to the plane,
- $d \in \mathbb{R}$ is the signed distance to the plane of the origin O, or equivalently $-\overrightarrow{OP} \,.\, \overrightarrow{n}$ for $P \in \mathcal{Pl}_{\overrightarrow{n},d}$.

Normalization

1. $\overrightarrow{n} \longleftarrow \overrightarrow{n} \,/\, \|\overrightarrow{n}\|$ (unit normal).

Initialization

A plane $\mathcal{Pl}_{\overrightarrow{n},d}$ is constructed from three non-collinear points P_1, P_2 and P_3 as follows:

1. $\overrightarrow{n} = \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} \,/\, \left\|\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}\right\|$
2. $d = -\overrightarrow{OP_1}.\overrightarrow{n}$

Point distance

The signed Euclidean distance of a point P to a plane $\mathcal{Pl}_{\overrightarrow{n},d}$ is:

$$\boxed{d(P, \mathcal{Pl}) = \overrightarrow{n} \,.\, \overrightarrow{OP} + d = \overrightarrow{n} \,.\, \overrightarrow{P_0P}}$$

for any point $P_0 \in \mathcal{Pl}$ (after the above normalization). The gradient of this distance *w.r.t.* P is:

$$\boxed{\nabla d(P, \mathcal{Pl}) = \overrightarrow{n}}$$

Least-squares fitting to a set of points

Least-squares fitting of a plane $\mathcal{Pl}^\star$ to a finite set of N points $\mathcal{P} = \{P_1, \ldots, P_N\}$ with respective weights $\{w_1, \ldots, w_N\}$ involves finding $\mathcal{Pl}^\star = \arg\min_{\overrightarrow{n},d} S(\overrightarrow{n}, d)$ with:

$$\boxed{S(\overrightarrow{n}, d) = \sum_{i=1}^{N} w_i \, d\left(P_i, \mathcal{Pl}_{\overrightarrow{n},d}\right)^2}$$

Introducing a Lagrangian multiplier $\lambda \in \mathbb{R}$ and modifying the objective function:

$$S'(\overrightarrow{n}, d, \lambda) \;=\; S(\overrightarrow{n}, d) - \lambda \left(\|\overrightarrow{n}\|^2 - 1 \right)$$

Looking for stationary points:

$$\begin{aligned}
\frac{\partial S'(\overrightarrow{n}, d, \lambda)}{\partial n_x} &= 2 \sum_{i=1}^{N} w_i \, P_{ix} \, d(P_i, \mathcal{Pl}) - 2 \lambda \, n_x \;=\; 0 \\
\frac{\partial S'(\overrightarrow{n}, d, \lambda)}{\partial n_y} &= 2 \sum_{i=1}^{N} w_i \, P_{iy} \, d(P_i, \mathcal{Pl}) - 2 \lambda \, n_y \;=\; 0 \\
\frac{\partial S'(\overrightarrow{n}, d, \lambda)}{\partial n_z} &= 2 \sum_{i=1}^{N} w_i \, P_{iz} \, d(P_i, \mathcal{Pl}) - 2 \lambda \, n_z \;=\; 0 \\
\frac{\partial S'(\overrightarrow{n}, d, \lambda)}{\partial d} &= 2 \sum_{i=1}^{N} w_i \, d(P_i, \mathcal{Pl}) \;=\; 0 \\
\frac{\partial S'(\overrightarrow{n}, d, \lambda)}{\partial \lambda} &= 1 - \|\overrightarrow{n}\|^2 \;=\; 0
\end{aligned}$$

leads to introducing the weighted centroid of the points:

$$\overline{P} = O + \frac{\sum_{i=1}^{N} w_i \, \overrightarrow{OP_i}}{\sum_{i=1}^{N} w_i}$$

which belongs to the plane since $\overrightarrow{n} \,.\, \overrightarrow{OP} + d = 0$ according to the equation from the fourth vanishing partial derivative.

Rewriting the objective function:

$$\sum_{i=1}^{N} w_i \, d(P_i, \mathcal{Pl})^2 \;=\; \sum_{i=1}^{N} w_i \left(\overrightarrow{n} \,.\, \overrightarrow{OP} + \overrightarrow{n} \,.\, \overrightarrow{\overline{P}P_i} + d \right)^2 \;=\; \overrightarrow{n} \,.\, (\mathrm{cov}(\mathcal{P}) \overrightarrow{n})$$

where $\mathrm{cov}(\mathcal{P}) = \sum_{i=1}^{N} w_i \, \overrightarrow{\overline{P}P} \otimes \overrightarrow{\overline{P}P}$ is the weighted covariance matrix of the set of weighted points $\mathcal{P}$.

The least-squares problem reduces to a simple eigenproblem that can be solved using Jacobi iterations for instance: the normal $\overrightarrow{n}$ is given by the eigenvector of $\mathrm{cov}(\mathcal{P})$ associated with the smallest eigenvalue and the distance d follows from this normal and the weighted centroid.

A.3 Sphere

Parametrization

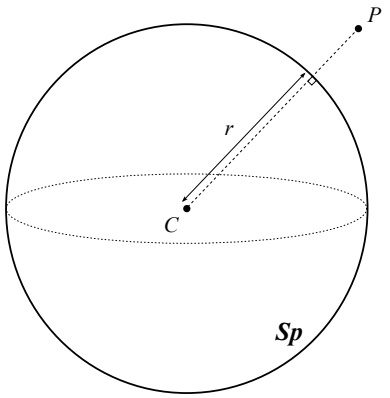


Figure A.2: *A sphere $\mathcal{Sp}_{C,r}$, the geometric meaning of its parameters and a point P.*

A sphere $\mathcal{Sp}_{C,r}$ is parametrized by C and r where (see Figure A.2):

- $C = (C_x, C_y, C_z) \in \mathbb{R}^3$ is the center of the sphere,
- $r \in \mathbb{R}^+$ is the radius of the sphere.

Normalization

None.

Initialization

A sphere $\mathcal{S}p_{C,r}$ is constructed from four non-coplanar points P_1, P_2, P_3 and P_4 as follows (with $P_i = (P_{ix}, P_{iy}, P_{iz})$ and $d_i = \left\|\overrightarrow{OP_i}\right\|$ for $i \in [\![1, 4]\!]$):

1. $C = \left(\frac{D_x}{2D_a}, \frac{D_y}{2D_a}, \frac{D_z}{2D_a}\right)$ is the circumcenter of the tetrahedron[1] whose vertices are the points P_1 P_2, P_3 and P_4 where:

$$D_x = \begin{vmatrix} d_1^2 & P_{1y} & P_{1z} & 1 \\ d_2^2 & P_{2y} & P_{2z} & 1 \\ d_3^2 & P_{3y} & P_{3z} & 1 \\ d_4^2 & P_{4y} & P_{4z} & 1 \end{vmatrix} \quad D_y = - \begin{vmatrix} d_1^2 & P_{1x} & P_{1z} & 1 \\ d_2^2 & P_{2x} & P_{2z} & 1 \\ d_3^2 & P_{3x} & P_{3z} & 1 \\ d_4^2 & P_{4x} & P_{4z} & 1 \end{vmatrix}$$

$$D_z = \begin{vmatrix} d_1^2 & P_{1x} & P_{1y} & 1 \\ d_2^2 & P_{2x} & P_{2y} & 1 \\ d_3^2 & P_{3x} & P_{3y} & 1 \\ d_4^2 & P_{4x} & P_{4y} & 1 \end{vmatrix}$$

$$D_a = \begin{vmatrix} P_{1x} & P_{1y} & P_{1z} & 1 \\ P_{2x} & P_{2y} & P_{2z} & 1 \\ P_{3x} & P_{3y} & P_{3z} & 1 \\ P_{4x} & P_{4y} & P_{4z} & 1 \end{vmatrix} \quad D_c = \begin{vmatrix} d_1^2 & P_{1x} & P_{1y} & P_{1z} \\ d_2^2 & P_{2x} & P_{2y} & P_{2z} \\ d_3^2 & P_{3x} & P_{3y} & P_{3z} \\ d_4^2 & P_{4x} & P_{4y} & P_{4z} \end{vmatrix}$$

2. $r = \frac{\sqrt{D_x^2 + D_y^2 + D_z^2 - 4D_aD_c}}{2|D_a|}$ is its radius.

Point distance

The signed distance of a point $P \in \mathbb{R}^3$ to a sphere $\mathcal{S}p_{C,r}$ is:

$$\boxed{d(P, \mathcal{S}p) = \left\|\overrightarrow{CP}\right\| - r}$$

The gradient of this distance *w.r.t.* P is:

$$\boxed{\nabla d(P, \mathcal{S}p) = \overrightarrow{CP} \,/\, \left\|\overrightarrow{CP}\right\|}$$

Least-squares fitting to a set of points

Least-squares fitting of a sphere $\mathcal{S}p^\star$ to a finite set of N points $\mathcal{P} = \{P_1, \ldots, P_N\}$ with respective weights $\{w_1, \ldots, w_N\}$ involves finding $\mathcal{S}p^\star = \arg\min_{C,r} S(C, r)$ with:

[1] A point $P = (P_x, P_y, P_z)$ belongs to the circumsphere of this tetradron iff $\begin{vmatrix} d^2 & P_x & P_y & P_z & 1 \\ d_1^2 & P_{1y} & P_{1y} & P_{1z} & 1 \\ d_2^2 & P_{2y} & P_{2y} & P_{2z} & 1 \\ d_3^2 & P_{3y} & P_{3y} & P_{3z} & 1 \\ d_4^2 & P_{4y} & P_{4y} & P_{4z} & 1 \end{vmatrix} = 0$ where $d = \left\|\overrightarrow{OP}\right\|$. Expanding this determinant leads to the formulae.

$$S(\overrightarrow{n}, d) = \sum_{i=1}^{N} w_i \, d\left(P_i, \mathcal{S}p_{C,r}\right)^2 = \sum_{i=1}^{N} w_i \, f_i(C,r)^2$$

Coefficients of the Jacobian matrix of $F(C,r) = \big(f_i(C,r)\big)$:

$$\begin{aligned}
\frac{\partial f_i(C,r)}{\partial C_x} &= -CP_{ix} \,/\, \left\|\overrightarrow{CP_i}\right\| \\
\frac{\partial f_i(C,r)}{\partial C_y} &= -CP_{iy} \,/\, \left\|\overrightarrow{CP_i}\right\| \\
\frac{\partial f_i(C,r)}{\partial C_z} &= -CP_{iz} \,/\, \left\|\overrightarrow{CP_i}\right\| \\
\frac{\partial f_i(C,r)}{\partial r} &= -1
\end{aligned}$$

A.4 Cylinder

Parametrization

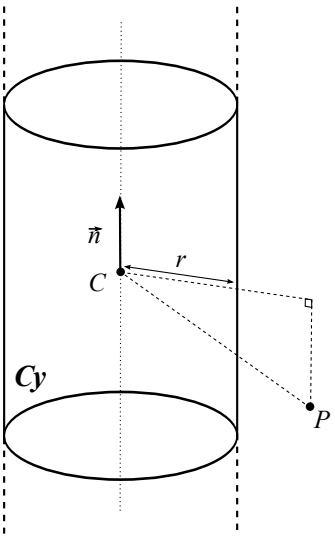


Figure A.3: *A cylinder $\mathcal{C}y_{C,\vec{n},r}$, the geometric meaning of its parameters and a point P.*

A cylinder $\mathcal{C}y_{C,\overrightarrow{n},r}$ is parametrized by C, $\overrightarrow{n}$ and r where (see Figure A.3):

- $C = (C_x, C_y, C_z) \in \mathbb{R}^3$ is a point on the axis of the cylinder,
- $\overrightarrow{n} = (n_x, n_y, n_z) \in \mathbb{R}^3$ is the axis direction,
- $r \in \mathbb{R}^+$ is the cylinder radius.

Normalization

1. $\overrightarrow{n} \longleftarrow\!\shortmid \overrightarrow{n} / \|\overrightarrow{n}\|$ (unit normal),
2. $C \longleftarrow\!\shortmid C - \left(\overrightarrow{n} \,.\, \overrightarrow{OC}\right) \overrightarrow{n}$ (closest point to the origin on the axis).

Initialization

A cylinder $\mathcal{C}y_{C,\overrightarrow{n},r}$ is constructed from three non-collinear points P_1, P_2 and P_3 as follows:

1. the axis direction $\overrightarrow{n}$ is chosen as the normal to the plane defined by the three points:
$$\overrightarrow{n} = \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} \,/\, \left\|\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}\right\|$$
2. the points P_1, P_2 and P_3 are projected on the plane defined by $\overrightarrow{n}$ and P_1 giving three points:
$$\begin{array}{rcccl} p_1 & = & (p_{1x}, p_{1y}, p_{1z}) & = & (0,0,0) \\ p_2 & = & (p_{2x}, p_{2y}, p_{2z}) & = & (\overrightarrow{P_1P_2} \,.\, \overrightarrow{u}, 0, 0) \\ p_3 & = & (p_{3x}, p_{3y}, p_{3z}) & = & (\overrightarrow{P_1P_3} \,.\, \overrightarrow{u}, \overrightarrow{P_1P_3} \,.\, \overrightarrow{v}, 0) \end{array}$$
in the orthonormal basis $(P_1, \overrightarrow{u}_x, \overrightarrow{u}_y, \overrightarrow{u}_z)$ with:
$$\begin{array}{rcl} \overrightarrow{u}_x & = & \overrightarrow{P_1P_2} \,/\, \left\|\overrightarrow{P_1P_2}\right\| \\ \overrightarrow{u}_y & = & \overrightarrow{n} \times \overrightarrow{P_1P_2} \,/\, \left\|\overrightarrow{n} \times \overrightarrow{P_1P_2}\right\| \\ \overrightarrow{u}_z & = & \overrightarrow{n} \end{array}$$
3. the center C and radius r of the cylinder derive from the center and radius of the circumcircle of the triangle with vertices p_1, p_2 and p_3. Noting $d_i = \left\|\overrightarrow{P_1p_i}\right\|$ for $i \in [\![1,3]\!]$:
$$d_x = \begin{vmatrix} d_1^2 & p_{1y} & 1 \\ d_2^2 & p_{2y} & 1 \\ d_3^2 & p_{3y} & 1 \end{vmatrix} = p_{3y} d_2^2 \qquad d_y = -\begin{vmatrix} d_1^2 & p_{1x} & 1 \\ d_2^2 & p_{2x} & 1 \\ d_3^2 & p_{3x} & 1 \end{vmatrix} = d_2^2 - d_3^2$$
$$d_a = \begin{vmatrix} p_{1x} & p_{1y} & 1 \\ p_{2x} & p_{2y} & 1 \\ p_{3x} & p_{3y} & 1 \end{vmatrix} = p_{2x} p_{3y} \qquad d_c = -\begin{vmatrix} d_1^2 & p_{1x} & p_{1y} \\ d_2^2 & p_{2x} & p_{2y} \\ d_3^2 & p_{3x} & p_{3y} \end{vmatrix} = 0$$
$$\begin{array}{rcl} C & = & P_1 + \dfrac{d_x}{2d_a} \overrightarrow{u}_x + \dfrac{d_y}{2d_a} \overrightarrow{u}_y \\ r & = & \dfrac{\sqrt{d_x^2 + d_y^2}}{2|d_a|} \end{array}$$

Point distance

The signed Euclidean distance of a point P to the cylinder $\mathcal{C}y_{C,\overrightarrow{n},r}$ is:

$$\boxed{d(P,\mathcal{C}y) = \left\|\overrightarrow{n} \times \overrightarrow{CP}\right\| - r = \left\|\overrightarrow{CP} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP}\right)\overrightarrow{n}\right\| - r}$$

(after the above normalization). The gradient of this distance *w.r.t.* P is:

$$\boxed{\nabla d(P,\mathcal{C}y) = \left(\overrightarrow{CP} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP}\right)\overrightarrow{n}\right) \,/\, \left\|\overrightarrow{n} \times \overrightarrow{CP}\right\|}$$

Least-squares fitting to a set of points

Least-squares fitting of a cylinder $\mathcal{C}y^\star$ to a finite set of N points $\mathcal{P} = \{P_1, \ldots, P_N\}$ with respective weights $\{w_1, \ldots, w_N\}$ involves finding $\mathcal{C}y^\star = \underset{C,\overrightarrow{n},r}{\arg\min}\, S(\overrightarrow{n}, r)$ with:

$$\boxed{S(\overrightarrow{n}, d) = \sum_{i=1}^{N} w_i \, d\left(P_i, \mathcal{C}y_{C,\overrightarrow{n},r}\right)^2 = \sum_{i=1}^{N} w_i \, f_i(C, \overrightarrow{n}, r)^2}$$

Coefficients of the Jacobian matrix of $F(C, \overrightarrow{n}, r) = \big(f_i(C, \overrightarrow{n}, r)\big)$:

$$\boxed{\begin{array}{rcl}
\dfrac{\partial f_i(C,\overrightarrow{n},r)}{\partial C_x} &=& -\left(CP_{ix} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) n_x\right) \,/\, \left\|\overrightarrow{n} \times \overrightarrow{CP_i}\right\| \\
\dfrac{\partial f_i(C,\overrightarrow{n},r)}{\partial C_y} &=& -\left(CP_{iy} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) n_y\right) \,/\, \left\|\overrightarrow{n} \times \overrightarrow{CP_i}\right\| \\
\dfrac{\partial f_i(C,\overrightarrow{n},r)}{\partial C_z} &=& -\left(CP_{iz} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) n_z\right) \,/\, \left\|\overrightarrow{n} \times \overrightarrow{CP_i}\right\| \\
\dfrac{\partial f_i(C,\overrightarrow{n},r)}{\partial n_x} &=& -\left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) \dfrac{CP_{ix} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) n_x}{\left\|\overrightarrow{n} \times \overrightarrow{CP_i}\right\|} \\
\dfrac{\partial f_i(C,\overrightarrow{n},r)}{\partial n_y} &=& -\left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) \dfrac{CP_{iy} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) n_y}{\left\|\overrightarrow{n} \times \overrightarrow{CP_i}\right\|} \\
\dfrac{\partial f_i(C,\overrightarrow{n},r)}{\partial n_z} &=& -\left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) \dfrac{CP_{iz} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP_i}\right) n_z}{\left\|\overrightarrow{n} \times \overrightarrow{CP_i}\right\|} \\
\dfrac{\partial f_i(C,\overrightarrow{n},r)}{\partial r} &=& -1
\end{array}}$$

A.5 Cone

Parametrization

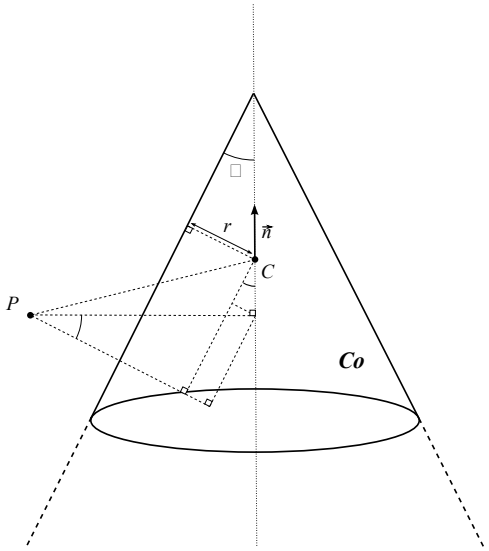


Figure A.4: *A cone $\mathcal{C}o_{\vec{n},d}$, the geometric meaning of its parameters and a point P.*

A cone $\mathcal{C}o_{C,\overrightarrow{n},d,\theta}$ is parametrized by C, $\overrightarrow{n}$, d and θ where (see Figure A.3):

- $C = (C_x, C_y, C_z) \in \mathbb{R}^3$ is a point on the cone axis (different from the apex),
- $\overrightarrow{n} = (n_x, n_y, n_z) \in \mathbb{R}^3$ is the axis direction ($\overrightarrow{n}$ pointing towards the apex),
- $d \in \mathbb{R}^+$ is the distance of C to the cone,
- $\theta \in [0, \frac{\pi}{2}]$ is the apex semi-angle.

Normalization

1. $\overrightarrow{n} \longleftarrow \overrightarrow{n} / \|\overrightarrow{n}\|$ (unit normal),
2. $C \longleftarrow C - \left(\overrightarrow{n} \,.\, \overrightarrow{OC}\right) \overrightarrow{n}$ (closest point to the origin on the axis),
3. $\theta \longleftarrow \theta \,[2\pi]$,
4. If $\theta > \pi$, $\theta \longleftarrow \theta[\pi]$ and $\overrightarrow{n} \longleftarrow -\overrightarrow{n}$,
5. If $\theta > \frac{\pi}{2}$, $\theta \longleftarrow \pi - \theta$,
6. If $d < 0$, $d \longleftarrow -d$ and $\overrightarrow{n} \longleftarrow -\overrightarrow{n}$.

Initialization

A cone $\mathcal{C}o_{C,\overrightarrow{n},d,\theta}$ is constructed from four non-coplanar points P_1, P_2, P_3 and P_4 as follows:

1. the axis direction $\overrightarrow{n}$ is chosen as the normal to the plane defined by the three points P_1, P_2 and P_3:

$$\overrightarrow{n} = \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} \,/\, \left\| \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} \right\|$$

2. the points P_1, P_2 and P_3 are projected on the plane defined by $\overrightarrow{n}$ and P_1 giving three points:

$$\begin{aligned} p_1 &= (p_{1x}, p_{1y}, p_{1z}) &&= (0,0,0) \\ p_2 &= (p_{2x}, p_{2y}, p_{2z}) &&= (\overrightarrow{P_1P_2} \,.\, \overrightarrow{u}, 0, 0) \\ p_3 &= (p_{3x}, p_{3y}, p_{3z}) &&= (\overrightarrow{P_1P_3} \,.\, \overrightarrow{u}, \overrightarrow{P_1P_3} \,.\, \overrightarrow{v}, 0) \end{aligned}$$

in the orthonormal basis $(P_1, \overrightarrow{u}_x, \overrightarrow{u}_y, \overrightarrow{u}_z)$ with:

$$\begin{aligned} \overrightarrow{u}_x &= \overrightarrow{P_1P_2} \,/\, \left\| \overrightarrow{P_1P_2} \right\| \\ \overrightarrow{u}_y &= \overrightarrow{n} \times \overrightarrow{P_1P_2} \,/\, \left\| \overrightarrow{n} \times \overrightarrow{P_1P_2} \right\| \\ \overrightarrow{u}_z &= \overrightarrow{n} \end{aligned}$$

3. the point C on the cone axis derives from the center of the circumcircle of the triangle with vertices p_1, p_2 and p_3. Noting $d_i = \left\| \overrightarrow{P_1p_i} \right\|$ for $i \in [\![1,3]\!]$:

$$d_x = \begin{vmatrix} d_1^2 & p_{1y} & 1 \\ d_2^2 & p_{2y} & 1 \\ d_3^2 & p_{3y} & 1 \end{vmatrix} = p_{3y} d_2^2 \qquad d_y = -\begin{vmatrix} d_1^2 & p_{1x} & 1 \\ d_2^2 & p_{2x} & 1 \\ d_3^2 & p_{3x} & 1 \end{vmatrix} = d_2^2 - d_3^2$$

$$d_a = \begin{vmatrix} p_{1x} & p_{1y} & 1 \\ p_{2x} & p_{2y} & 1 \\ p_{3x} & p_{3y} & 1 \end{vmatrix} = p_{2x} p_{3y} \qquad d_c = -\begin{vmatrix} d_1^2 & p_{1x} & p_{1y} \\ d_2^2 & p_{2x} & p_{2y} \\ d_3^2 & p_{3x} & p_{3y} \end{vmatrix} = 0$$

$$\begin{aligned} C &= P_1 + \frac{d_x}{2d_a} \overrightarrow{u}_x + \frac{d_y}{2d_a} \overrightarrow{u}_y \\ r &= \frac{\sqrt{d_x^2 + d_y^2}}{2|d_a|} \end{aligned}$$

4. the axis direction $\overrightarrow{n}$ is (optionally) corrected: $\overrightarrow{n}$ should points towards the cone apex defined to be on the same side of the P_1, P_2, P_3 plane as P_4 if P_4 is inside the cylinder $\mathcal{C}y_{C,\overrightarrow{n},r}$ and on the other side in the opposite case,

5. d is the distance of C to the line defined by P_4 and the projection of P_4 on the circle:

$$\begin{aligned} d_0 &= \overrightarrow{n} \,.\, \overrightarrow{CP_4} \\ \overrightarrow{u}_0 &= \overrightarrow{n} \times \overrightarrow{CP_4} \\ r_0 &= \|u_0\| \\ P_5 &= C + \overrightarrow{u}_0 + d_0 \, \overrightarrow{n} \\ \overrightarrow{u} &= ((r_0 - r) \, \overrightarrow{u}_0 / r_0 + d_0 \, \overrightarrow{n}) \, / \, \|(r_0 - r) \, \overrightarrow{u}_0 / r_0 + d_0 \, \overrightarrow{n}\| \end{aligned}$$

$$d = \left\| \overrightarrow{u} \times \overrightarrow{CP_5} \right\|$$

6. the apex semi-angle θ is deduced from d and r:

$$\theta = \cos^{-1} \frac{d}{r}$$

Point distance

The signed Euclidean distance of a point P to the cone $\mathcal{C}o_{C,\overrightarrow{n},d,\theta}$ is:

$$\boxed{d(P, \mathcal{C}o) = \left\| \overrightarrow{n} \times \overrightarrow{CP} \right\| \cos\theta + \left(\overrightarrow{n} \,.\, \overrightarrow{CP} \right) \sin\theta - d}$$

(after the above normalization). The gradient of this distance *w.r.t.* P is:

$$\boxed{\nabla d(P, \mathcal{C}o) = \left(\overrightarrow{CP} - \left(\overrightarrow{n} \,.\, \overrightarrow{CP} \right) \overrightarrow{n} \right) / \left\| \overrightarrow{n} \times \overrightarrow{CP} \right\| \cos\theta + \overrightarrow{n} \sin\theta}$$

Least-squares fitting to a set of points

Least-squares fitting of a cone $\mathcal{C}o^\star$ to a finite set of N points $\mathcal{P} = \{P_1, \ldots, P_N\}$ with respective weights $\{w_1, \ldots, w_N\}$ involves finding $\mathcal{C}o^\star = \underset{C,\overrightarrow{n},d,\theta}{\arg\min} \, S(C, \overrightarrow{n}, d, \theta)$ with:

$$\boxed{S(C, \overrightarrow{n}, d, \theta) = \sum_{i=1}^{N} w_i \, d\left(P_i, \mathcal{C}o_{C,\overrightarrow{n},d,\theta}\right)^2 = \sum_{i=1}^{N} w_i \, f_i(C, \overrightarrow{n}, d, \theta)^2}$$

Coefficients of the Jacobian matrix of $F(C, \overrightarrow{n}, d, \theta) = \left(f_i(C, \overrightarrow{n}, d, \theta)\right)$:

$$
\begin{aligned}
\frac{\partial f_i(C,\overrightarrow{n},d,\theta)}{\partial C_x} &= -\left(\frac{CP_{ix}-\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_x}{\left\|\overrightarrow{n}\times\overrightarrow{CP_i}\right\|}\cos\theta+n_x\sin\theta\right)\\
\frac{\partial f_i(C,\overrightarrow{n},d,\theta)}{\partial C_y} &= -\left(\frac{CP_{iy}-\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_y}{\left\|\overrightarrow{n}\times\overrightarrow{CP_i}\right\|}\cos\theta+n_y\sin\theta\right)\\
\frac{\partial f_i(C,\overrightarrow{n},d,\theta)}{\partial C_z} &= -\left(\frac{CP_{iz}-\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_z}{\left\|\overrightarrow{n}\times\overrightarrow{CP_i}\right\|}\cos\theta+n_z\sin\theta\right)\\
\frac{\partial f_i(C,\overrightarrow{n},d,\theta)}{\partial n_x} &= -\left(\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)\frac{CP_{ix}-\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_x}{\left\|\overrightarrow{n}\times\overrightarrow{CP_i}\right\|}\cos\theta\right.\\
&\qquad\left.+\left(\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_x-CP_{ix}\right)\sin\theta\right)\\
\frac{\partial f_i(C,\overrightarrow{n},d,\theta)}{\partial n_y} &= -\left(\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)\frac{CP_{iy}-\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_y}{\left\|\overrightarrow{n}\times\overrightarrow{CP_i}\right\|}\cos\theta\right.\\
&\qquad\left.+\left(\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_y-CP_{iy}\right)\sin\theta\right)\\
\frac{\partial f_i(C,\overrightarrow{n},d,\theta)}{\partial n_z} &= -\left(\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)\frac{CP_{iz}-\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_z}{\left\|\overrightarrow{n}\times\overrightarrow{CP_i}\right\|}\cos\theta\right.\\
&\qquad\left.+\left(\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)n_z-CP_{iz}\right)\sin\theta\right)\\
\frac{\partial f_i(C,\overrightarrow{n},d,\theta)}{\partial d} &= -1\\
\frac{\partial f_i(C,\overrightarrow{n},d,\theta)}{\partial \theta} &= -\left(\left\|\overrightarrow{n}\times\overrightarrow{CP_i}\right\|\sin\theta-\left(\overrightarrow{n}\,.\,\overrightarrow{CP_i}\right)\cos\theta\right)
\end{aligned}
$$

APPENDIX B

Cell complexes and geometric queries

This chapter describes in detail the exact or approximate computation of the various cell complexes used throughout this thesis. The implementation of the main geometric queries used in our surface fitting framework, namely point location and traversal along a segment are also discussed.

Contents

The two geometric queries required by the visibility term presented in Chapter 2 are first defined.

Definition (Point location query)**.** *The point location query* LOCATE(P) *on a d-complex maps a point $P \in \mathbb{E}^d$ to a cell C such that $P \cap C \neq \emptyset$ (the cell C contains P).*

The result of this query is always defined in this thesis since the domain of our cell complexes is the whole ambient space $\mathbb{E}^3$.

Definition (Line walk query)**.** *A line walk* LINEWALK(P, Q) *on a d-complex maps two points $P, Q \in \mathbb{E}^d$ to a sequence of cells $(C_1, \ldots, C_N)$ traversed by the line segment $[PQ]$ with $P \in C_1$ and $Q \in C_Q$.*

A few remarks:

- the line walk query always begins by locating P, if possible this location can also be provided to avoid an additional query.
- a reverse line walk can be used to find the cells "behind" a point, with a trivial modification to stop either after the first encountered facet (Chapters 2, 4, 7 and 8) or after some distance along the line segment (Chapter 3).
- for simplicity, we assume that the line segment $[PQ]$ intersect only facets.
- the data structure used to represent the complex is supposed to allow efficient retrieval of the incident faces so that the facets crossed by the line can be recovered from two adjacent cells.

B.1 Delaunay triangulations

Definition

Let $\mathcal{P} = \{P_1, \ldots, P_N\}$ be a set of N points in $\mathbb{E}^d$. The Voronoi cell associated to a point P_i, denoted by $\mathrm{Vor}(P_i)$, is the region of space that is closer from P_i than from all other points in $\mathcal{P}$:

$$\mathrm{Vor}(P_i) = \left\{ P \in \mathbb{E}^d \ : \ \forall\, j \neq i \ \ \|P - P_i\| \leq \|P - P_j\| \right\}$$

$\mathrm{Vor}(P_i)$ is the intersection of $N - 1$ half-spaces bounded by the bisector planes of segments $[P_i P_j]$ with $j \neq i$. $\mathrm{Vor}(P_i)$ is therefore a convex polytope, possibly unbounded. The Voronoi diagram of $\mathcal{P}$, denoted by $\mathrm{Vor}(\mathcal{P})$, is the cell complex induced by the Voronoi cells $\mathrm{Vor}(P_i)$.

The Delaunay triangulation $\mathrm{Del}(\mathcal{P})$ of $\mathcal{P}$ is defined as the geometric dual of the Voronoi diagram: there is an edge between two points P_i and P_j in the Delaunay triangulation if and only if their Voronoi cells $\mathrm{Vor}(P_i)$ and $\mathrm{Vor}(P_j)$ have a non-empty intersection. It yields a triangulation of $\mathcal{P}$, that is to say a partition of the convex hull of $\mathcal{P}$ into d-dimensional simplices (*i.e.* into

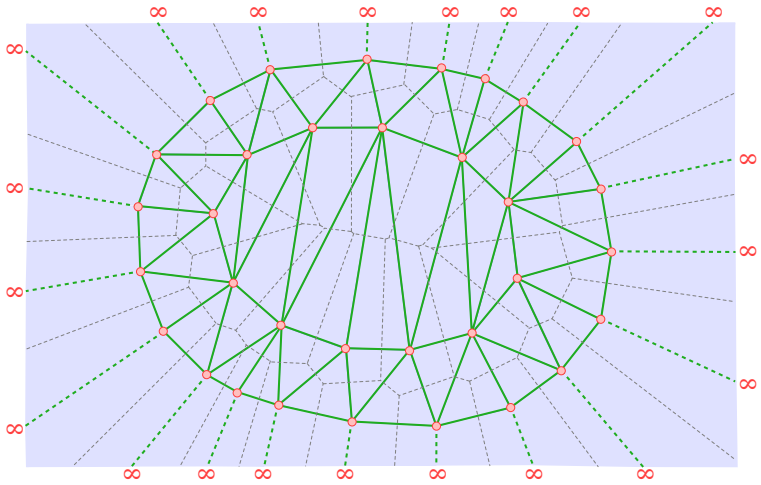


Figure B.1: ***Delaunay triangulation and Voronoi diagram*** *of a set of points in 2D. The vertices of the triangulation correspond to the input points plus the infinite vertices. As in Chapter 2, vertices are in red, edges in green and cells (triangles here) in blue. The edges of the associated Voronoi diagram are superimposed in dashed gray.*

triangles in 2D, into tetrahedra in 3D, and so on). Figure B.1 displays an example of a Voronoi diagram and its associated Delaunay triangulation in the plane. An "infinite" vertex and infinite tetrahedra (one for each facet of the convex hull of $\mathcal{P}$) are added to the triangulation so that $\mathrm{dom}(\mathrm{Del}(\mathcal{P})) = \mathbb{E}^d$.

The algorithmic complexity of the Delaunay triangulation of N points is at most $\mathcal{O}(N \log N)$ in 2D, and $\mathcal{O}(N^2)$ in 3D or generally $O(N \log N + N^{\lceil d/2 \rceil})$ and its combinatorial complexity is $\mathcal{O}(N^{\lceil d/2 \rceil})$ in the general case [Boissonnat and Yvinec, 1998a].

Construction

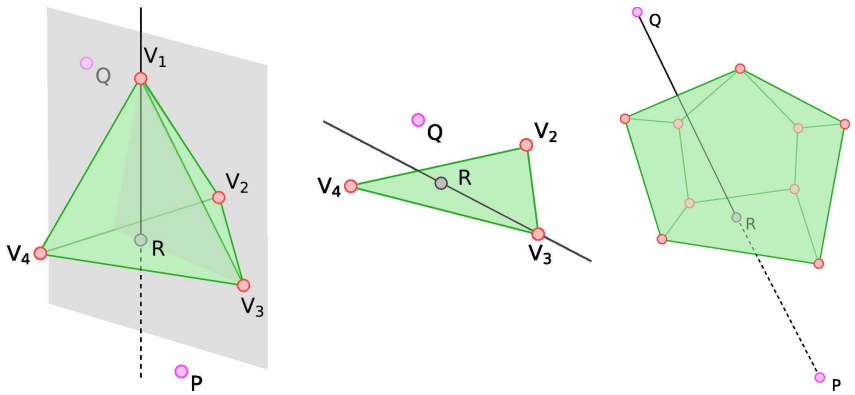
The Computational Geometry Algorithms Library (CGAL) [cga, 2009] provides an excellent implementation of the algorithm described in [Devillers, 2002] to compute the Delaunay triangulation of 3D points. It is robust to degenerate configurations and floating-point error thanks to the use of symbolic perturbation and exact geometric predicates while still being able to process millions of points per minute on a standard workstation as demonstrated in Chapters 3 and 4.

Point location

The algorithm of [Devillers, 2002] actually computes a hierarchy of Delaunay triangulations of a decreasing sequence of subsets of the input point cloud. This allows the location of a query point to be efficiently found: starting from the highest level of the hierarchy, the closest vertex is located in the current level and the search proceeds in the next level of the hierarchy.

Line walk

The only query lacking from CGAL is the line walk in 3D. For a line walk query LineWalk(P, Q), once a tetrahedron C_1 containing the point P is located, the facet F intersected by the ray $[PQ)$ and incident to the second tetrahedron C_2 of the sequence of tetrahedra can be located by brute force. Then, the intersection point R of F with $[PQ)$ is determined. As shown in Figure B.2(a), if V_1 is the vertex opposite to F and V_2, V_3 and V_4 the other vertices of C_2, the outgoing facet in C_2 can be found from the location of Q *w.r.t.* the planes passing supported by $\{R, V_1, V_2\}$, $\{R, V_1, V_3\}$ and $\{R, V_1, V_4\}$. Once this new outgoing facet is found, a new intersection point can be computed and the search continues until the point Q lies in the tetrahedron. This line walk avoids the explicit computation of ray intersections with all the facets of traversed tetrahedron and is not specific to Delaunay triangulations.



(a) Line walk in a triangulation.

(b) Line walk in a polyhedral complex.

Figure B.2: ***Walking along a line in complexes.***

B.2 BSP polyhedral complexes

Construction

We first consider a special case of polyhedral complexes, the arrangement of hyperplanes and recall its definition:

Definition (Hyperplanes arrangement)**.** *Let $\mathcal{PL}$ be a set of n hyperplanes in $\mathbb{E}^d$. The intersection of a finite number of half-spaces is a bounded or unbounded (convex) polytope, and so $\mathcal{PL}$ induces a decomposition of $\mathbb{E}^d$ into a collection of bounded or unbounded polytopes with pairwise disjoint interiors. These polytopes and their faces (in 3D, their vertices, edges and facets) form a pure cell complex of dimension d which is called the* arrangement *of $\mathcal{PL}$. This cell complex is denoted by* $\mathrm{Arr}(\mathcal{PL})$.

An arrangement in $\mathbb{E}^d$ is called a d-arrangement. In the sequel, for simplicity, we will only consider sets of hyperplanes in *general position*: any subset of d planes meets at a point and the intersection of $d+1$ planes is empty. The corresponding arrangement is called a *simple* arrangement. The total number of faces in a simple arrangement[1] of n hyperplanes is $\Theta(n^d)$.

If we consider the d-arrangement $\mathrm{Arr}(\mathcal{PL})$ and $\mathcal{Pl}$ a plane not belonging to $\mathcal{PL}$, the complex formed by the d-faces in $\mathrm{Arr}(\mathcal{PL})$ and their faces is called the zone of $\mathcal{Pl}$ in the arrangement $\mathrm{Arr}(\mathcal{PL})$. The complexity of any zone[2] in the d-arrangement $\mathrm{Arr}(\mathcal{PL})$ of n hyperplanes is $\Theta(n^{d-1})$.

Based on this property, algorithms exist to incrementally construct an arrangement in time $\Theta(n^d)$. [Edelsbrunner, 1987] provides a detailed explanation of an algorithm explicitly computing the incidence graph[3] of all the faces of the arrangement. In [Boissonnat and Yvinec, 1998b], a shorter description of a similar algorithm computing only the incidence graph of the 0, 1 and 2-faces is given.

Algorithm. 4 gives a very summarized description of the algorithm from [Edelsbrunner, 1987]. The line 4.2 computes the simple d-arrangement of the first d planes (in 3D, it constructs 1 vertex, 6 edges, 12 faces and 8 cells). Each remaining plane is then inserted in the current arrangement. First, an edge intersecting the plane is found, then the zone of the plane in the current arrangement is identified (first 0 and 1-faces, then higher dimensional faces) and finally, faces of the zone that requires splitting are split and the incidence graph is updated. Of course, the description given is highly simplified, it is however sufficient to understand the only modification required to compute polyhedral complexes induced by BSP. For complete details on the original arrangement computation algorithm, we refer the reader to [Edelsbrunner, 1987].

[1] It is still $\mathcal{O}(n^d)$ in a non-simple arrangement.
[2] It is still $\mathcal{O}(n^{d-1})$ in a non-simple arrangement.
[3] The data structure of this graph also stores a point at each node, this point is located

Algorithm 4 An overview of the incremental algorithm of [Edelsbrunner, 1987] to compute the incidence graph $\mathcal{I}$ of an arrangement of n planes $\mathcal{P}\ell_1, \ldots, \mathcal{P}\ell_n$

1: **function** BUILDARRANGEMENT($\mathcal{P}\ell_1, \ldots, \mathcal{P}\ell_n$)
Require: $n \geq d$
2: $\mathcal{I} \leftarrow$ BUILDTRIVIALARRANGEMENT($\mathcal{P}\ell_1, \ldots, \mathcal{P}\ell_d$)
3: **for** $i \leftarrow d+1, n$ **do**
4: Pick a random vertex V in the arrangement $\mathrm{Arr}(\{\mathcal{P}\ell_1, \ldots, \mathcal{P}\ell_{i-1}\})$
5: Find an edge E incident to V such that $\mathrm{aff}(V) \cap \mathcal{P}\ell_i \neq \emptyset$
6: Walk along $\mathrm{aff}(V)$ to find an edge E_0 such that $E_0 \cap \mathcal{P}\ell_i \neq \emptyset$
7: Starting from E_0, mark all 0- and 1-faces of the zone of $\mathcal{P}\ell_i$
8: Mark all the k-faces of the zone of $\mathcal{P}\ell_i$, $k \in [\![2, d]\!]$
9: Replace the marked faces of $\mathrm{Arr}(\{\mathcal{P}\ell_1, \ldots, \mathcal{P}\ell_{i-1}\})$ and update the incidence graph $\mathcal{I}$ accordingly
10: **end for**
11: **end function**

The algorithm is slightly modified to compute the polyhedral complex induced by a binary space partitioning described by a tree. A bounding box surrounds the finite cells of the complex. The arrangement corresponding to the $2d$ planes of this bounding box is computed using this algorithm[4]. Then, by traversing the BSP tree from the root (the root corresponds to the only bounded cell among the 3^d cells of the initial arrangement), the polyhedral complex is incrementally built as follows: each inner node of the tree corresponds to a cell of the current polyhedral complex that should be split by the corresponding plane. This split is performed by restricting the action of lines 4.7, 4.8 and 4.9 to the current cell: only k-faces within the cell are considered, as if the zone of the plane was restricted to the current cell. This restriction ensures that only faces relevant for the BSP complex are created. The complexity of the modified algorithm naturally follows the complexity of the BSP complex. Furthermore, in Chapter 6, several cells can be split simultaneously by a single plane: the zone of this plane can be restricted to all these cells to achieve the construction as shown in Figure B.3.

CGAL [cga, 2009] does provide constructions and queries for 2D arrangements of lines (generic enough to handle bounded and unbounded line segments and curves). It however does not provides an implementation of arrangements of planes in 3D, so the algorithm described in [Edelsbrunner, 1987] and summarized here was implemented along with the above modification to compute polyhedral complexes induced by BSPs. The involved queries (point location and line walk traversal) to be described in the next

in the interior of the k-face corresponding to the node.

[4] Modified to handle non-simple arrangements.

sections were also implemented, all this made possible and transparently robust to degenerate configurations and floating-point error thanks to the excellent CGAL kernel which provides not only exact predicates but also constructions[5].

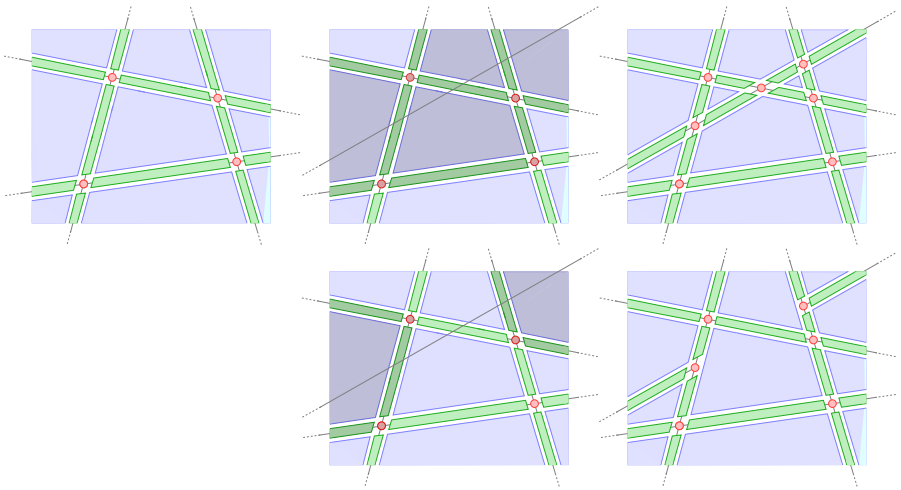


Figure B.3: ***From hyperplanes arrangements to BSP polyhedral complexes.*** *First row: an arrangement of 4 lines in the planes, the zone of a line in this arrangement, the arrangement of the 5 lines. Second row: the zone of a line restricted to a set of cells and the BSP complex.*

Point location

To locate a cell containing a point in a polyhedral complex, different strategies are possible: - exhaustive search of all the complex cells, - walk along a line from a known cell by casting a ray to the query point and traversing the complex, - put landmarks of known location in the complex, use a k-tree to find the nearest landmark and locate with a line walk, - decompose the cells of the complex into simpler cells of constant complexity and use a search structure. As a compromise between implementation time and efficiency, a slight variant of the "walking along a line" is used. Indeed, pinhole or push-broom cameras have finite projection centers and the cells of these centers may be located in advance or tracked. The line walk required by the surface fitting can then directly be used from these cells.

[5]Exact constructions are required because the location of created vertices is used by the incremental construction.

Line walk

The line walk query on polyhedral complexes follows the line walk algorithm in triangulations of Section B.1. One major difference however is the fact that the simplified intersection test cannot be used anymore since cells can be general d-polytopes and not d-simplices with only $d+1$ vertices. The search for an outgoing facet has to exhaustively check all the facets of a crossed polytopial cell for intersection (see Figure B.2(b). This is done by first testing for intersecting with the supporting plane of a facet, and then ensuring that the intersection lies inside the polygon of the facet.

B.3 Generalized BSP complexes

As noted in Chapter 8, the practical exact and robust computation of complexes whose cells are delimited by higher order surfaces (even when restricted to second order surfaces only) is still the subject of active research.

Construction

A practical solution consists in using a tetrahedral meshing of the space refined to approximate the recovered shapes. To compute this approximation of the cell complex induced by the generalized BSP tree, the adaptive volume mesh generator of [Pons et al., 2007b] is employed. This mesh generator extends the surface meshing algorithm of [Boissonnat and Oudot, 2005] to multi-label partitions. The underlying algorithm works by refining a 3D Delaunay triangulation and only requires as input an oracle answering for a point which label it is associated with. In our case, this oracle answers the leaf of the generalized BSP tree where the point is located. The output is a labeled Delaunay triangulation, which approximates the interfaces between the distinct valued domains: each tetrahedron of this triangulation is labeled according to its associated leaf. From this labeled simplicial complex, we recover an approximating complex whose cells are union of simplicial cells and facets are union of simplicial facets. These cells of our approximated BSP induced complex are found as the connected components of identically labeled tetrahedra. The facets of the complex (and the whole adjacency graph) are found as the connected components of triangle facets between two tetrahedra with the same labels pairs. The mapping between the simplicial cells and the computed cells of the approximation is maintained as well as the correspondence between simplicial facets and facets of the approximated complex.

Point location

The point location is answered using the approximating triangulation as in Section B.1 and the found tetrahedron is mapped to the corresponding cell

of the BSP complex.

Line walk

The line walk query follows Section B.1 with modifications to account for cells and facets of the approximated complex.

Post-processing

The actual implementation of [Boissonnat and Oudot, 2005] only accurately meshes smooth surfaces while piecewise-smooth surfaces are considered in Chapter 8. The implementation design of the algorithm in CGAL [Rinaud and Yvinec, 2007] is flexible and would allow the inclusion of specific refinement criteria for sharp features if available. Such refinement criteria were not available at the time. Instead, in Chapter 8, the output surface and in particular the boundaries and vertices shared between shapes, is improved by an additional post-processing step: all the vertices of the triangulation are iteratively reprojected on each of the shape regions to which they belong.

Finally, an optional edge collapse simplification following [Lindstrom and Turk, 1998] with a cost accounting for normal deviation may be applied to simplify the surface mesh while taking into account for its known geometry.

APPENDIX C

Publications of the author

Journal papers

- Patrick Labatut, Jean-Philippe Pons, Renaud Keriven and Vu Hoang Hiep. Global and visibility-consistent dense multi-view stereo for large-scale scenes. In preparation.

- Patrick Labatut, Jean-Philippe Pons and Renaud Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*. To appear.

Conference and workshop papers

- Patrick Labatut, Renaud Keriven and Jean-Philippe Pons. Fast level set multi-view stereo on graphics hardware. In *3rd International Symposium on 3D Data Processing, Visualization and Transmission*, pages 774–781, 2006.

- Patrick Labatut, Jean-Philippe Pons and Renaud Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and graph cuts. In *IEEE International Conference on Computer Vision*, 2007.

- Vu Hoang Hiep, Renaud Keriven, Patrick Labatut and Jean-Philippe Pons. Towards high-resolution large-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1437, 2009.

- Patrick Labatut, Jean-Philippe Pons and Renaud Keriven. Hierarchical shape-based surface reconstruction for dense multi-view stereo. *The 2009 IEEE International Workshop on 3-D Digital Imaging and Modeling*, 2009.

Bibliography

Autodesk ImageModeler, 2009. http://www.autodesk.com/imagemodeler. 91

CGAL, Computational Geometry Algorithms Library, 2009. http://www.cgal.org/. 173, 176

Google SketchUp, 2009. http://sketchup.google.com/. 91

MINPACK, 1999. http://www.netlib.org/minpack/. 159

Stanford 3D scanning repository, 2009. http://graphics.stanford.edu/data/3Dscanrep/. 9

Ehsan Aganj, Jean-Philippe Pons, and Renaud Keriven. Globally optimal spatio-temporal reconstruction from cluttered videos. In *Asian Conference on Computer Vision*, 2009. 148

Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Cláudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, 2003. 23

Rémi Allègre, Raphaëlle Chaine, and Samir Akkouche. A dynamic surface reconstruction framework for large unstructured point sets. In *Symposium on Point-Based Graphics*, pages 17–26, 2006. 147

Rémi Allègre, Raphaëlle Chaine, and Samir Akkouche. A streaming algorithm for surface reconstruction. In *Symposium on Geometry Processing*, pages 79–88, July 2007. 60, 147

Cédric Allène, Jean-Philippe Pons, and Renaud Keriven. Seamless image-based texture atlases using multi-band blending. In *International Conference on Pattern Recognition*, 2008. 72, 75

Pierre Alliez, David Cohen-Steiner, Yiying Tong, and Mathieu Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *Symposium on Geometry Processing*, pages 39–48, July 2007. 26

Nina Amenta and Marshall Bern. Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry*, 22(4):481–504, 1999. 25, 46

Nina Amenta, Marshall Bern, and David Eppstein. The crust and the beta-skeleton: Combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60(2):125–135, 1998a. 52

Nina Amenta, Marshall Bern, and Manolis Kamvysselisy. A new Voronoi-based surface reconstruction algorithm. In *ACM SIGGRAPH*, pages 415–421, 1998b. 25

Nina Amenta, Sunghee Choi, and Ravi Kolluri. The power crust. In *ACM Symposium on Solid Modeling and Applications*, pages 249–260, 2001. 25, 58

Nina Amenta, Sunghee Choi, Tamal K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *International Journal of Computational Geometry and Applications*, 12(1–2):125–141, 2002. 25

Ben Appleton and Hugues Talbot. Globally minimal surfaces by continuous maximal flows. *IEEE Transactions on Pattern Analysis and Machince Intelligence*, 28:106–118, 2006. 33

Dominique Attali, Jean-Daniel Boissonnat, and André Lieutier. Complexity of the Delaunay triangulation of points on surfaces: The smooth case. In *ACM Symposium on Computational Geometry*, pages 237–246, 2003. 46

Caroline Baillard and Andrew Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 559–565, 1999. 92

Dana H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981. 103

Atsuhiko Banno, Tomohito Masuda, Takeshi Oishi, and Katsushi Ikeuchi. Flying laser range sensor for large-scale site-modeling and its applications in Bayon digital archival project. *International Journal of Computer Vision*, 78(2–3):207–222, 2008. 11, 68, 147

Adrien Bartoli. A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding*, 105(1):42–59, 2007. 91

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. 16, 110

Fausto Bernardini, Joshua Mittleman, Holly E. Rushmeier, Cláudio T. Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4): 349–359, 1999. 25

Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machince Intelligence*, 14(2): 239–256, 1992. 9

Jean-Daniel Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, 1984. 25

Jean-Daniel Boissonnat and Frédéric Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry*, 22(1–3):185–203, 2002. 26

Jean-Daniel Boissonnat and Steve Oudot. Provably good sampling and meshing of surfaces. *Graphical Models*, 67(5):405–451, 2005. 132, 178, 179

Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry*, chapter Voronoi diagrams: Euclidean metric, Delaunay complexes, pages 435–443. Cambridge University Press, 1998a. 173

Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry*, chapter Arrangements of hyperplanes, pages 335–365. Cambridge University Press, 1998b. 175

Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, and Mariette Yvinec. Triangulations in CGAL. In *ACM Symposium on Computational Geometry*, pages 11–18, 2000. 53

Matthew Bolitho, Michael Kazhdan, Randal Burns, and Hugues Hoppe. Multilevel streaming for out-of-core surface reconstruction. In *Symposium on Geometry Processing*, pages 69–78, July 2007. 60, 147

Yuri Boykov and Vladimir Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *IEEE International Conference on Computer Vision*, pages 26–33, 2003. 24, 33, 34

Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004. 33, 53

Yuri Boykov and Victor Lempitsky. From photohulls to photoflux optimization. In *British Machine Vision Conference*, volume 3, pages 1149–1158, 2006. 69

Derek Bradley, Tamy Boubekeur, and Wolfgang Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 26

Adrian Broadhurst, Tom W. Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *IEEE International Conference on Computer Vision*, pages 388–393, 2001. 69

Matthew Brown and David G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *International Conference on 3-D Digital Imaging and Modeling (3DIM 2005)*, June 2005. 11

Neill D. F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision*, volume 5302 of *Lecture Notes in Computer Science*, pages 766–779, 2008. 70

Rodrigo L. Carceroni and Kiriakos N. Kutulakos. Multi-view scene capture by surfel sampling. *International Journal of Computer Vision*, 49(2):175–214, 2002. 148

Jonathan C. Carr, Richard K. Beatson, Jon B. Cherrie, Tim J. Mitchell, W. Richard Fright, Bruce C. McCallum, and Tim R. Evans. Reconstruction and representation for 3D objects with radial basis functions. In *ACM SIGGRAPH*, pages 67–76, 2001. 23

Frédéric Cazals and Joachim Giesen. *Effective Computational Geometry for Curves and Surfaces*, chapter Delaunay Triangulation Based Surface Reconstruction, pages 231–276. Mathematics and Visualization. Springer, 2006. 25

Raphaëlle Chaine. A geometric convection approach of 3-D reconstruction. In *Symposium on Geometry Processing*, pages 218–229, June 2003. 25, 26

Haifeng Chen, Peter Meer, and David E. Tyler. Robust regression for data with multiple structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1069–1075, 2001. 109

Jie Chen and Baoquan Chen. Architectural modeling from sparsely scanned range data. *International Journal of Computer Vision*, 78(2–3):223–236, July 2008. 93

Fan R. Chung. *Spectral Graph Theory*. AMS, 1997. 32

David Cohen-Steiner, Éric Colin de Verdière, and Mariette Yvinec. Conforming delaunay triangulations in 3d. In *Symposium on Computational Geometry*, pages 199–208, 2002. 136

David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. In *ACM SIGGRAPH*, pages 905–914, 2004. 94

Dorin Comaniciu and Peter Meer. Mean Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 24(5):603–619, 2002. 103

Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. The variable bandwidth mean shift and data-driven scale selection. In *International Conference on Computer Vision*, pages 438–445, 2001. 104

William J. Cook, William H. Cunningham, William R. Pulleyblank, , and Alexander Schrijver. *Combinatorial Optimization*, chapter Maximum Flow Problems. John Wiley & Sons, 1998. 33

Nico Cornelis, Bastian Leibe, Kurt Cornelis, and Luc Van Gool. 3D urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2–3):121–141, 2007. 91

James M. Coughlan and A. L. Yuille. Manhattan world: orientation and outlier detection by bayesian inference. *Neural Computation*, 15:1063–1088, May 2003. 92

Jérôme Courchay, Jean-Philippe Pons, Pascal Monasse, and Renaud Keriven. Dense and accurate spatio-temporal multi-view stereovision. In *Asian Conference on Computer Vision*, 2009. 148

Brian Curless. *New Methods For Surface Reconstruction From Range Images*. PhD thesis, Computer Graphics Laboratory, Stanford University, 1997. 10, 11

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *ACM SIGGRAPH*, pages 303–312, 1996. 2, 22, 36, 48, 57

Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs. In *ACM SIGGRAPH*, pages 11–20, 1996. 91

Xavier Décoret, Frédo Durand, François X. Sillion, and Julie Dorsey. Billboard clouds for extreme model simplification. In *ACM SIGGRAPH*, pages 689–696, 2003. 103

Amaël Delaunoy, Emmanuel Prados, Pau Gargallo, Jean-Philippe Pons, and Peter Sturm. Minimizing the multi-view stereo reprojection error for triangular surface meshes. In *British Machine Vision Conference*, 2008. 70, 77, 147

Alain Dervieux and François Thomasset. A finite element method for the simulation of Rayleigh-Taylor instability. *Lecture Notes in Mathematics*, 771:145–159, 1979. 24

Frédéric Devernay. C/C++ MINPACK, 2009. `http://devernay.free.fr/hacks/cminpack.html`. 160

Olivier Devillers. The Delaunay hierarchy. *International Journal of Foundations of Computer Science*, 13(2):163–180, 2002. 173, 174

Tamal K. Dey and Samrat Goswami. Tight cocone: A water-tight surface reconstructor. In *ACM Symposium on Solid Modeling and Applications*, pages 127–134, 2003. 25

Tamal K. Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry: Theory and Applications*, 35(1): 124–141, 2006. 25, 58

Tamal K. Dey, Gang Li, and Jian Sun. Normal estimation for point clouds: A comparison study for a Voronoi based method. In *Symposium on Point-Based Graphics*, pages 39–46, 2005. 58

Anthony Dick, Philip H. S. Torr, and Roberto Cipolla. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2):111–134, 2004. 90

E. A. Dinic. Algorithm for solution of a problem of maximum flow in networks with power estimation. *Soviet Math. Dokl.*, 11:1277–1280, 1970. 32

Ye Duan, Liu Yang, Hong Qin, and Dimitris Samaras. Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. In *European Conference on Computer Vision*, volume 3, pages 238–251, 2004. 70, 77

Richard O. Duda and Peter E. Hart. Use of the Hough transform to detect lines and circles in pictures. *Communication of the ACM*, 15(1):11–15, 1972. 101

Laurent Dupont, Michael Hemmer, Sylvain Petitjean, and Elmar Schömer. Complete, exact and efficient implementation for computing the adjacency graph of an arrangment of quadrics. In *15th Annual European Symposium on Algorithms*, pages 633–644, 2007. 132

Ilya Eckstein, Jean-Philippe Pons, Yiying Tong, C. C. Jay Kuo, and Mathieu Desbrun. Generalized surface flows for mesh processing. In *Symposium on Geometry Processing*, pages 183–192, 2007. 77

Hebert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13:43–72, 1994. 25

Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*, chapter Constructing Arrangements, pages 121–137. Springer, 1987. 155, 175, 176

Lixin Fan and Timo Pylvänäinen. Robust scale estimation from ensemble inlier sets for random sample consensus methods. In *European Conference on Computer Vision*, number 3, pages 182–195, 2008. 148

Olivier Faugeras and Renaud Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998. 2, 69, 70, 74, 76, 77

Olivier Faugeras, Elizabeth Lebras-Mehlman, and Jean-Daniel Boissonnat. Representing stereo data with the Delaunay triangulation. *Artificial Intelligence Journal*, 44(1–2):41—-87, 1990. 73

Olivier Faugeras, Quang-Tuan Luong, and Theodore Papadopoulou. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, 2001. 11, 12

Raphael A. Finkel and Jon Louis Bentley. Quad Trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974. 39

Valentino Fiorin, Paolo Cignoni, and Roberto Scopigno. Out-of-core MLS reconstruction. In *International Conference on Computer Graphics and Imaging*, pages 27–34, 2007. 60, 147

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 91, 105, 109, 155

Shachar Fleishman, Daniel Cohen-Or, and Claudio T. Silva. Robust moving least-squares fitting with sharp features. In *ACM SIGGRAPH*, volume 24, pages 544–552, July 2005. 23

Efi Fogel, Dan Halperin, Lutz Kettner, Monique Teillaud, Ron Wein, and Nicola Wolpert. Arrangements. In Jean-Daniel Boissonnat and Monique Teillaud, editors, *Effective Computational Geometry for Curves and Surfaces*, Mathematics and Visualization, pages 1–66. Springer, 2006. 132

Lester Randolph Ford and Delbert Ray Fulkerson. *Flows in Networks*. Princeton University Press, 1962. 32

Richard Franke and Gregory Nielson. Smooth interpolation of large sets of scattered data. volume 15, pages 1691–1704, 1980. 23

Friedrich Fraundorfer, Konrad Schindler, and Horst Bischof. Piecewise planar scene reconstruction from sparse correspondences. *Image and Vision Computing*, 24(4):395–406, 2006. 91

Daniel Freedman and Petros Drineas. Energy minimization via graph cuts: Settling what is possible. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 33

Henry Fuchs, Zvi M. Kedem, and Bruce F. Naylor. On visible surface generation by a priori tree structures. In *International Conference on Computer Graphics and Interactive Techniques*, pages 124–133, 1980. 96, 118

Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. In *European Conference on Computer Vision*, pages 564–577, May 2006. 69

Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007. 2, 18, 26, 70, 79, 83, 84, 85, 92

Yasutaka Furukawa and Jean Ponce. Dense 3D motion capture from synchronized video streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008. 148

Yasutaka Furukawa and Jean Ponce. Dense 3D motion capture for human faces. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2009. 148

Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Manhattan-world stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 92

David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007. 91

Pau Gargallo and Peter Sturm. Bayesian 3D modeling from images using multiple depth maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 885–891, June 2005. 70

Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *ACM SIGGRAPH*, pages 209–216, 1997. 93, 94

Nicola Geismann, Michael Hemmer, and Elmar Schömer. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! In *17th Annual Symposium on Computational Geometry*, pages 264–273, 2001. 132

Natasha Gelfand and Leonidas J. Guibas. Shape segmentation using local slippage analysis. In *Symposium on Geometry Processing*, pages 214–223, July 2004. 92

Jan-Mark Geusebroek, Gertjan J. Burghouts, and Arnold W. M. Smeulders. The Amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005. 123

Michael Goesele, Briand Curless, and Steven M. Seitz. Multi-view stereo revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2402–2409, 2006. 70

Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *IEEE International Conference on Computer Vision*, 2007. 2, 18, 26, 70, 147

Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940, October 1988. 32

D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B, Methodological*, 51(2):271–279, 1989. 33

André Guéziec, Gabriel Taubin, Francis Lazarus, and William Horn. Cutting and stitching: Converting sets of polygons to manifold surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):136–151, 2001. 39

Martin Habbecke and Leif Kobbelt. Iterative multi-view plane fitting. In *Vision, Modeling, and Visualization*, 2006. 91

Chris Harris and Mike Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, 1988. 14

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 11, 16

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002. 28

Carlos Hernández and Francis Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding, special issue on 'Model-based and image-based 3D Scene Representation for Interactive Visualization'*, 96(3):367–392, December 2004. 69, 70, 77

Carlos Hernández, George Vogiatzis, and Roberto Cipolla. Probabilistic visibility for multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007. 70

Adrian Hilton. Scene modelling from sparse 3D data. *Image and Vision Computing*, 23(10):900–920, 2005. 92

Adam Hoover, Gillian Jean-Baptiste, Xiaoyi Jiang, Patrick J. Flynn, Horst Bunke, Dmitry B. Goldgof, Kevin Bowyer, David W. Eggert, Andrew Fitzgibbon, and Robert B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996. 92

Hugues Hoppe. Progressive meshes. In *ACM SIGGRAPH*, pages 99–108, 1996. 93

Hugues Hoppe. View-dependent refinement of progressive meshes. In *ACM SIGGRAPH*, pages 189–198, 1997. 93

Hugues Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *IEEE Visualization*, pages 59–66, 1999. 93

Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *ACM SIGGRAPH*, pages 71–78, 1992. 22

Alexander Hornung and Leif Kobbelt. Robust and efficient photo-consistency estimation for volumetric 3D reconstruction. In *European Conference on Computer Vision*, 2006a. 69

Alexander Hornung and Leif Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 503–510, 2006b. 69, 70

Alexander Hornung and Leif Kobbelt. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *Symposium on Geometry Processing*, pages 41–50, June 2006c. 24, 33

Paul V. C. Hough. Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation*, 1959. 101

Paul V. C. Hough. Methods and means for recognising complex patterns, U.S. Patent 3069654, December 1962. 101

Peter J. Huber. *Robust Statistics*. Wiley, 1981. 101

Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *IEEE International Conference on Computer Vision*, October 2007. 148

John Illingworth and Josef Kittler. The adaptive Hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):690–698, 1989. 103

Michal Jancosek and Tomás Pajdla. Segmentation based multi-view stereo. In *Computer Vision Winter Workshop*, 2009. 83

Hailin Jin, Anthony J. Yezzi, Yen-Hsi Tsai, Li-Tien Cheng, and Stefano Soatto. Estimation of 3D surface shape and smooth radiance from 2D images: A level set approach. *Journal of Scientific Computing*, 19:267–292, 2003. 147

Hailin Jin, Stefano Soatto, and Anthony J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63(3):175–189, 2005. 69, 70, 77, 147

Hailin Jin, Daniel Cremers, Dejun Wang, Emmanuel Prados, Anthony J. Yezzi, and Stefano Soatto. 3-d reconstruction of shaded objects from multiple images under unknown illumination. *International Journal of Computer Vision*, 3:245–256, 2008. 147

Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM SIGGRAPH*, volume 26, 2007. 18, 19, 57

Michael Kazhdan. Reconstruction of solid models from oriented point sets. In *Symposium on Geometry Processing*, pages 73–82, July 2005. 23, 24

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, pages 61–70, June 2006. 2, 23, 24, 26, 56, 57, 58, 60, 70, 121, 122, 124, 137, 153

Danil Kirsanov and Steven J. Gortler. A discrete global minimization algorithm for continuous variational problems. Technical Report TR-14-04, Harvard Computer Science, July 2004. 34

Nahum Kiryati, Yuval Eldar, and Alfred M. Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24(4):303–316, 1991. 103

Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *International Conference on Computer Graphics and Interactive Techniques*, pages 105–114, 1998. 77

Leif Kobbelt, Mario Botsch, Ulrich Schwanecke, and Hans-Peter Seidel. Feature sensitive surface extraction from volume data. In *ACM SIGGRAPH*, pages 57–66, 2001. 22, 93

Kalin Kolev, Maria Klodt, Thomas Brox, and Daniel Cremers. Continuous global optimization in multiview 3D reconstruction. *International Journal of Computer Vision*, 2009. 69

Ravi Krishna Kolluri. Provably good moving least squares. *ACM Transactions on Algorithms*, 4(2), 2008. 23

Ravikrishna Kolluri, Jonathan R. Shewchuk, and James F. O'Brien. Spectral surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing*, pages 11–21, July 2004. 25, 32, 52, 58, 59

Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, volume 3, pages 82–96, May 2002. 70

Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004. 33

Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. 2, 69

Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and graph cuts. In *IEEE International Conference on Computer Vision*, October 2007. 5, 12, 34, 74, 156

Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*, 2009a. to appear. 5, 156

Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Hierarchical shape-based surface reconstruction for dense multi-view stereo. In *The 2009 IEEE International Workshop on 3-D Digital Imaging and Modeling*, 2009b. 5, 156

Florent Lafarge, Renaud Keriven, and Mathieu Brédif. Combining meshes and geometric primitives for accurate and semantic modeling. In *British Machine Vision Conference*, 2009. 135

Peter Lancaster and Kestusis Salkauskas. Surfaces generated by moving least-squares methods. *Mathematics of Computation*, 37:141–158, 1981. 23

Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994. 68

Kil-Moo Lee, Peter Meer, and Rae-Hong Park. Robust adaptive segmentation of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:200–205, 1998. 105

Victor Lempitsky and Yuri Boykov. Global optimization for shape fitting. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007. 24, 34, 38, 93

Victor Lempitsky, Yuri Boykov, and Denis Ivanov. Oriented visibility for multiview reconstruction. In *European Conference on Computer Vision*, volume 3, pages 226–238, May 2006. 70

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 2:164–168, 1944. 159

David Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*. Springer, 2003. 23

Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean E. Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital Michelangelo project: 3D scanning of large statues. In *ACM SIGGRAPH*, pages 131–144, 2000. 147

Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005. 70, 77

Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998. 14

Peter Lindstrom and Greg Turk. Image-driven mesh optimization. Technical Report GIT-GVU-00-16, Georgia Institute of Technology, 2000. 93

Peter Lindstrom and Greg Turk. Fast and memory efficient polygonal simplification. In *IEEE Visualization*, volume 544, pages 279–286, October 1998. 93, 179

Stuart P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. 94

Hervé Lombaert, Yiyong Sun, Leo Grady, and Chenyang Xu. A multilevel banded graph cuts method for fast image segmentation. In *IEEE International Conference on Computer Vision*, pages 259–265, October 2005. 33

William E. Lorensen and Harvey E. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. In *ACM SIGGRAPH*, pages 163–169, 1987. 22

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 12

David Luebke, Martin Reddy, Jonathan D. Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner. *Level of Detail for 3D Graphics*. Morgan Kaufmann, 2002. 93

Donald Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441, 1963. 159

Daniel Martinec. *Robust Multiview Reconstruction*. PhD thesis, Czech Technical University, June 2008. 11

Boris Mederos, Nina Amenta, Luiz Velho, and Luiz Henrique de Figueiredo. Surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing*, pages 53–62, July 2005. 25

Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *IEEE International Conference on Computer Vision*, October 2007. 70

James V. Miller and Charles V. Stewart. MUSE: Robust surface fitting using unbiased scale estimates. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 300–306, 1996. 105

Bojan Mohar. *The Laplacian spectrum of graphs*, volume 2, pages 871–898. Wiley, 1991. 31

Bojan Mohar. Some applications of laplace eigenvalues of graphs. In *Graph Symmetry: Algebraic Methods and Applications*, volume 497 of *NATO ASI Series C*, pages 227–275. Kluwer, 1997. 31

Jorge J. Moré, Burton S. Garbow, and Kenneth E. Hillstrom. User guide for MINPACK-1. Technical Report ANL-80-74, Argonne National Laboratory, 1980. 159

Jorge J. Moré, Danny C. Sorensen, Kenneth E. Hillstrom, and Burton S. Garbow. *Sources and Development of Mathematical Software*, chapter The MINPACK Project, pages 88–111. Prentice-Hall, 1984. 159

T. M. Murali and Thomas A. Funkhouser. Consistent solid and boundary representations from arbitrary polygonal data. In *Symposium on Interactive 3D Graphics*, 1997. 119

D. R. Myatt, Philip H. S. Torr, Slawomir J. Nasuto, J. Mark Bishop, and R. Craddock. NAPSAC: High noise, high dimensional robust estimation - It's in the bag. In *British Machine Vision Conference*, 2002. 113

Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. In *ACM SIGGRAPH*, volume 22, pages 463–470, 2003. 23, 58

Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Shape Modeling International*, pages 153–164, 2004. 23

Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988. 24

Emmanuel Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962. 103

Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004. 11

Marc Pollefeys, David Nistér, Jan-Michael Frahm, Amir Akbarzadeh, Philippos Mordoha, Brian Clipp, Christopher Engels, David Gallup, Seon Joo Kim, Paul Merrell, Christina Salmi, Sudipta N. Sinha, Brad Talton, Liang Wang, Qingxiong Yang, Henrik Stewénius, Ruigang Yang, Greg Welch, and Herman Towles. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2–3):143–167, 2008. 147

Jean-Philippe Pons and Jean-Daniel Boissonnat. Delaunay deformable models: Topology-adaptive meshes based on the restricted Delaunay triangulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007. 77

Jean-Philippe Pons, Renaud Keriven, and Olivier Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193, 2007a. 2, 69, 70, 75, 76, 77, 148, 154

Jean-Philippe Pons, Florent Ségonne, Jean-Daniel Boissonnat, Laurent Rineau, Mariette Yvinec, and Renaud Keriven. High-quality consistent meshing of multi-label datasets. In *Information Processing in Medical Imaging*, pages 198–210, 2007b. 155, 178

John Princen, John Illingworth, and Josef Kittler. A hierarchical approach to line extraction based on the Hough transform. *Computer Vision, Graphics, and Image Processing*, 52(1):52–77, 1990. 103

Laurent Rinaud and Mariette Yvinec. A generic software design for Delaunay refinement meshing. *Computational Geometry*, 38(1–2):100–110, 2007. 179

Jorma Rissanen. Modeling by the shortest data description. *Automatica*, 14: 465–471, 1978. 148

Peter J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984. 105

Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. 1987. 105

Marie Samozino, Marc Alexa, Pierre Alliez, and Mariette Yvinec. Reconstruction with voronoi centered radial basis functions. In *Symposium on Geometry Processing*, pages 51–60, 2006. 26

Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007. 92

Ruwen Schnabel, Patrick Degener, and Reinhard Klein. Completion and reconstruction with primitive shapes. In *EUROGRAPHICS*, 2009. 93

Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, November 1999. 69

Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Rick Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 519–526, June 2006. `http://vision.middlebury.edu/mview/`. 2, 9, 68, 69, 79, 90, 153

Craig M. Shakarji. Least-squares fitting algorithms of the NIST algorithm testing system. *National Institute of Standards and Technology Journal of Research*, 103(6):633–641, 1998. 157

Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008. 94

Andrei Sharf, Thomas Lewiner, Ariel Shamir, Leif Kobbelt, and Daniel Cohen-Or. Competing fronts for coarse-to-fine surface reconstruction. In *EUROGRAPHICS*, volume 25, pages 389–398, 2006. 24

Chen Shen, James F. O'Brien, and Jonathan R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *ACM SIGGRAPH*, volume 23, pages 896–904, 2004. 23

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 888–905, 2000. 31

Hang Si. TetGen: A quality tetrahedral mesh generator and three-dimensional Delaunay triangulator, April 2007. `http://tetgen.berlios.de/`. 136

Hang Si and Klaus Gaertner. Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations. In *14th International Meshing Roundtable*, pages 147–163, September 2005. 136

Sudipta N. Sinha and Marc Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *IEEE International Conference on Computer Vision*, pages 349–356, October 2005. 69

Sudipta N. Sinha, Philippos Mordohai, and Marc Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *IEEE International Conference on Computer Vision*, October 2007. 34, 70

Sudipta N. Sinha, Drew Steedly, Richard Szeliski, Maneesh Agrawala, and Marc Pollefeys. Interactive 3D architectural modeling from unordered photo collections. In *ACM SIGGRAPH Asia*, 2008. 91

Greg Slabaugh and Gozde Unal. Active polyhedron: Surface evolution theory applied to deformable meshes. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 84–91, 2005. 77

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008. 11, 147

Dan Snow, Paul Viola, and Ramin Zabih. Exact voxel occupancy with graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2000. 2

Jonathan Starck and Adrian Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007. 148

Jonathan Starck, Gregor Miller, and Adrian Hilton. Volumetric stereo with silhouette and feature constraints. In *British Machine Vision Conference*, volume 3, pages 1189–1198, 2006. 69, 70

H. Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci. C1. III*, pages 801–804, 1956. 31, 32, 94

Charles V. Stewart. MINPRAN: A new robust estimator for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10): 925–938, 1995. 105

Charles V. Stewart. Bias in robust estimation caused by discontinuities and multiple structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):818–833, 1997. 106

Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997. 31

Christoph Strecha, Tinne Tuytelaars, and Luc Van Gool. Dense matching of multiple wide-baseline views. In *IEEE International Conference on Computer Vision*, volume 2, pages 1194–1201, 2003. 70

Christoph Strecha, Rik Fransens, and Luc Van Gool. Wide-baseline stereo from multiple views: A probabilistic account. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 552–559, 2004. 70, 83, 84, 85

Christoph Strecha, Rik Fransens, and Luc Van Gool. Combined depth and outlier estimation in multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2394–2401, 2006. 2, 70, 83

Christoph Strecha, Wolfgang von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. `http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html`. 9, 13, 17, 18, 68, 70, 71, 79, 81, 82, 127, 154

Raghav Subbarao and Peter Meer. Projection based m-estimators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009. 104

Robert Endre Tarjan. *Data Structures and Network Algorithms*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1983. 33

Gabriel Taubin. Nonplanar curve and surface estimation in 3-space. In *IEEE Conference on Robotics and Automation*, 1988. 113

Gabriel Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991. 113

Gabriel Taubin. An improved algorithm for algebraic curve and surface fitting. In *IEEE International Conference on Computer Vision*, 1993. 113

Gabriel Taubin. A signal processing approach to fair surface design. In *ACM SIGGRAPH*, pages 351–358, 1995. 77

Roberto Toldo and Andrea Fusiello. Robust multiple structures estimation with J-Linkage. In *European Conference on Computer Vision*, volume 5302 of *Lecture Notes in Computer Science*, pages 537–547, October 2008. 109, 148

Philip H. S. Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000. 106, 109, 110

Son Tran and Larry Davis. 3D surface reconstruction using graph cuts with surface constraints. In *European Conference on Computer Vision*, volume 2, pages 219–231, 2006. 69, 70

Adrien Treuille, Aaron Hertzmann, and Steven M. Seitz. Example-based stereo with general BRDFs. In *European Conference on Computer Vision*, volume 2, pages 457–469, 2004. 69

Bill Triggs, P. McLauchlan, Richard Hartley, and A. Fitzgibbon". Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372, 2000. 11

Greg Turk and Marc Levoy. Zippered polygon meshes from range images. pages 311–318, 1994. 36

Radim Tylecek and Radim Sara. Depth map fusion with camera position refinement. In *Computer Vision Winter Workshop*, 2009. 70, 79, 83, 84

Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward, and Philip H. S. Torr. VideoTrace: Rapid interactive scene modelling from video. In *ACM SIGGRAPH*, volume 26, pages 86–90, 2007. 91

Sundar Vedula, Simon Baker, Steven M. Seitz, and Takeo Kanade. Shape and motion carving in 6D. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2000. 147

Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):475–480, March 2005. 148

George Vogiatzis, Philip H. S. Torr, and Roberto Cipolla. Multi-view stereo via volumetric graph-cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 391–398, 2005. 2, 69

George Vogiatzis, Carlos Hernández, Phil H. S. Torr, and Roberto Cipolla. Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2241–2246, December 2007. 34, 69

J. Vollmer, Robert Mencl, and Heinrich Müller. Improved Laplacian smoothing of noisy surface meshes. *Computer Graphics Forum*, 8(3):131–138, 1999. 53

Hoang Hiep Vu, Renaud Keriven, Patrick Labatut, and Jean-Philippe Pons. Towards high-resolution large-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2009. `http://imagine.enpc.fr/demos/stereo/`. 5, 156

C. S. Wallace and D. M. Boulton. An information measure for classification. *Computer Journal*, 11(2):185–194, August 1968. 148

Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995. 58

Tomas Werner and Andrew Zisserman. New techniques for automated architectural reconstruction from photographs. In *European Conference on Computer Vision*, 2002a. 90

Tomas Werner and Andrew Zisserman. Model selection for automated architectural reconstruction from multiple views. In *British Machine Vision Conference*, 2002b. 90

Ross T. Whitaker. A level set approach to 3D reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998. 24, 53

M. Kass A. Witkin and Dimitri Terzopoulos. Snakes: Active contour models. *Internation Journal of Computer Vision*, 1(4):321–331, January 1988. 24

Jianhua Wu and Leif Kobbelt. Structure recovery via hybrid variational surface approximation. In *EUROGRAPHICS*, volume 24, pages 277–284, 2005. 95

Lei Xu, Erkki Oja, and Pekka Kultanen. A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5): 331–338, 1990. 103

Dong-Ming Yan, Yang Liu, and Wenping Wang. Quadric surface extraction by variational shape approximation. In *Geometric Modeling and Processing*, 2006. 95

Ruigang Yang and Marc Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 211–220, 2003. 15

Ruigang Yang, Marc Pollefeys, and Greg Welch. Dealing with textureless regions and specular highlights: A progressive space carving scheme using a novel photo-consistency measure. In *IEEE International Conference on Computer Vision*, volume 1, pages 576–584, 2003. 69

Tianli Yu, Narendra Ahuja, and Wei-Chao Chen. SDG cut: 3D reconstruction of non-lambertian objects using graph cuts on surface distance grid. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2269–2276, 2006. 69

Christopher Zach, Thomas Pock, and Horst Bischof. A globally optimal algorithm for robust TV-L1 range image integration. In *IEEE International Conference on Computer Vision*, October 2007. 70

Andrei Zaharescu, Edmond Boyer, and Radu P. Horaud. TransforMesh: a topology-adaptive mesh-based approach to surface evolution. In *Asian Conference on Computer Vision*, pages 166–175, November 2007. 77, 83, 84

Andrei Zaharescu, Cédric Cagniart, Slobodan Ilic, Edmond Boyer, and Radu P. Horaud. Camera clustering for multi-resolution 3-D surface reconstruction. In *ECCV 2008 Workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008. 147

Wei Zhang and Jana Kosecká. Nonparametric estimation of multiple structures with outliers. In *ECCV 2006 Workshop on Dynamical Vision*, pages 60–74, 2006. 109

Hong Kai Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 194–201, 2001. 24, 25, 26, 53

Marco Zuliani, Charles S. Kenney, and B. S. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *International Conference on Image Processing*, number 3, pages 153–156, 2005. 109